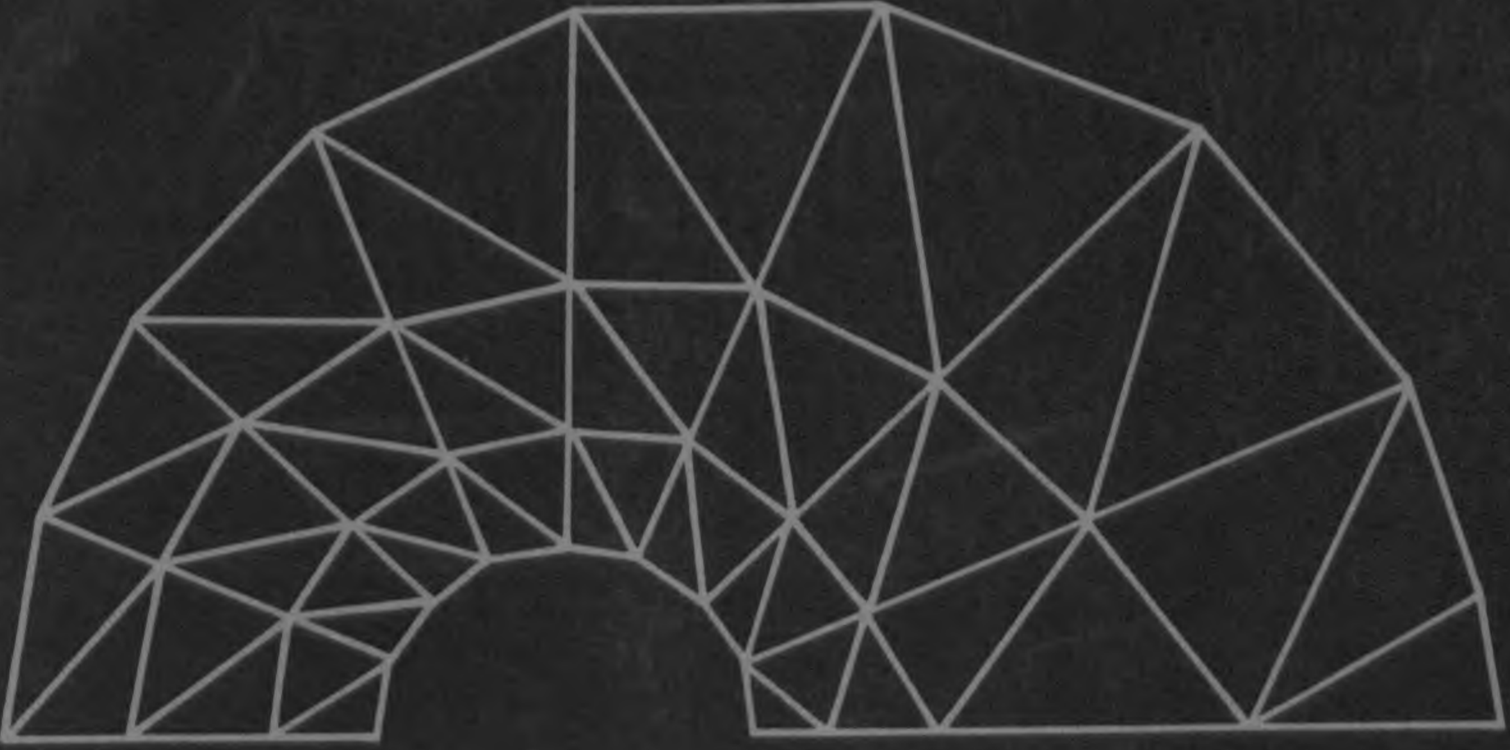
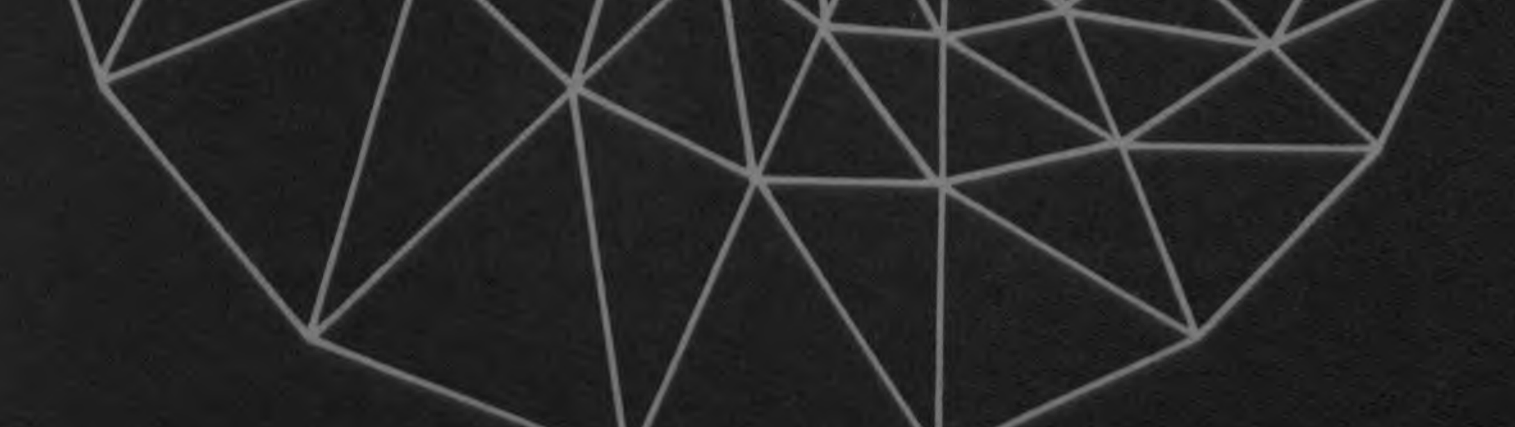


APPLIED FINITE ELEMENT ANALYSIS

An Apple II Implementation

J. ROBERT COOKE · DENNY C. DAVIS

Includes BASIC Source Code for All Programs



APPLIED FINITE ELEMENT ANALYSIS
An Apple II Implementation

p324 l 38
p341 l 22
344 l 60
355 l 200
383 l 300

APPLIED FINITE ELEMENT ANALYSIS
An Apple II Implementation

J. ROBERT COOKE
Cornell University

DENNY C. DAVIS
Washington State University

JOHN WILEY & SONS
New York • Chichester • Brisbane • Toronto • Singapore

ISBN 0-471-82337-6

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

PREFACE

The finite element method is an enormously important tool for engineering analysis, and familiarity with the method is rapidly becoming an expected part of engineering curricula. The swift expansion in the availability and power of digital computers with low cost interactive graphics could enhance the popularity of this numerical technique as well as provide a vehicle for instruction in its use. In this book we demonstrate the power of the finite element methodology when used with microcomputers even though their graphics resolution and memory are limited.

We have developed a fully interactive, graphics-based implementation for the Apple II family of microcomputers. With this set of programs you can satisfactorily solve two-dimensional and axisymmetric steady state heat conduction, electrostatics, ideal fluid flow, seepage, and elasticity problems of the complexity presented in Segerlind's *Applied Finite Element Analysis* (1984).

In addition, the interactive graphics interface allows you to focus on the intellectual content of the methodology rather than on the considerable bookkeeping details. To enable you to thoroughly explore the algorithms, we have provided unprotected diskettes and annotated BASIC source code.

Many individuals have contributed to the evolution of this book. E.T. Sobel participated in the writing of the Applesoft code, R. Gates and J.Y. Lee solved the illustrative problems and, thereby, exposed many bugs and "other opportunities" for improvement of the program. J. Booker, W. Coats, G. Elliott, R. Gustafson, J. Nieber, P. Parchomchuk, R. Rand, J. Reid, R. Stroshine, and S. Upadhyaya deserve special thanks for providing constructive comments and suggestions. We also benefited from the suggestions of our students at Cornell and Washington State.

We thank L. Segerlind and R. Gustafson for granting us permission to use the examples from their books, and K. King for his encouragement with this software development effort. Computerland of Ithaca made an Apple II-C available to us so that we might resolve compatibility problems. We thank Bill Stenquist of John Wiley for his willingness to experiment with this approach to the distribution of educational software.

Several people participated in the preparation of the manuscript. N. Cooke improved the presentation of the material and proofread the text. B. Czarniecki entered much of the material on an IBM PC. S. Bates coordinated the telecommunication from the Apple II and IBM PC to the Philips Micom word processor and from the word processor to Graphic Arts. R. Wiiki coordinated the conversion to the Epics Composition System for output on the Comp/Edit Digital Phototypesetter. C. Benzon scheduled the typesetting process and S. MacKay gave careful attention to the design and layout of the book.

We spent hundreds of hours testing these programs, but "opportunities for improvement" inevitably remain. Although we will be interested to know of your experiences with these programs (and hope that our considerable effort has been justified), we will be unable to personally provide user support.

Ithaca	J. Robert Cooke
Pullman	Denny C. Davis

CONTENTS

3.0 **ELASTICITY**

4.0 **PROGRAMMER'S GUIDE**

1.0 INTRODUCTION

1.1 The Finite Element Method

The finite element method derives much of its appeal from the scope of problems which it can handle in a systematic manner. In this section we present only a brief overview of the methodology. After you have gained an appreciation of the procedural aspects of the method using the software included with this book, your subsequent study of the theoretical foundations can proceed in a more meaningful manner. For further discussion of the fundamentals of the method, refer to Segerlind's *Applied Finite Element Analysis* (1984) or Cook's *Concepts and Applications of Finite Elements* (1981).

Figure 1.1 illustrates the range and complexity of problems that may be solved with this Apple II implementation. In each example we assume that you formulated a boundary value problem and stated the relevant assumptions explicitly. To formulate a boundary value problem you must specify the governing partial differential equation and the boundary conditions. Finding the solution to the partial differential equation consists of calculating values at various points throughout the interior of the region. You can derive assorted other quantities from the solution.

Rather than trying to find the solution at all interior points, you find the solution at a finite number of interior points. You can approximate the value at any other point through an interpolation process. The finite set of interior points that you choose defines a finite number of subregions or elements which collectively approximate the original region geometry. Figure 1.2 illustrates the methodology.

Throughout the present book we restrict our discussion to triangular elements, the simplest two-dimensional element. The three vertices or nodes completely define the element. You can also treat the special three-dimensional class of problems having symmetry about an axis, i.e., bodies of revolution, using annular elements of triangular cross-section. A general purpose program would provide a library of elements, not just the simplest two utilized here.

The creation of the mesh of elements to approximate the entire region requires judgment and creativity as well as tedious calculations. Well-shaped (e.g., equilateral) triangles produce more accurate results than very slender ones. Triangles should be smaller in regions where change in the calculated value (temperature or displacement) is expected to be greatest, because linear interpolation is used to approximate values within the interior of the triangle.

Once the geometry has been approximated by a mesh of elements (i.e., triangles), you can specify the material properties for each of the elements. In the third step - specification of the conditions to be satisfied along the boundary - you complete the problem formulation.

You can now convert the problem into a system of simultaneous linear algebraic equations; the unknowns correspond to the values (e.g., displacement components or temperature) at the vertices of the triangles. You solve the system of equations using Gaussian elimination. In addition to finding the unknowns, you compute various derived quantities (e.g., gradients, stresses and strains). You can present the extensive output for a problem in tabular form. However, a graphical representation of the results is often more meaningful.

The finite element method allows the mathematical details to move into the background. You need not be explicitly concerned with the governing partial differential equations. Similarly, you need not be a programmer. Problem formulation consists solely of data input, unlike alternatives such as the finite difference method which usually requires source code changes to reflect different conditions.

You may wonder why we bothered to implement the finite element method on a small microcomputer since it is a method usually reserved for larger mainframe or, at least, super minicomputers. The answer lies in the graphics of the microcomputer. The computational details of even small instructional finite element problems require the use of a digital computer. Even on a mainframe the process is tedious, and the student needs diligence to get past the procedural details. Consequently, we feel that the low cost graphics of the microcomputer can greatly enhance the learning process of the student.

With graphics, much of the bookkeeping detail moves into the background. For example, you

Fig. 1.1 Illustrative boundary value problems

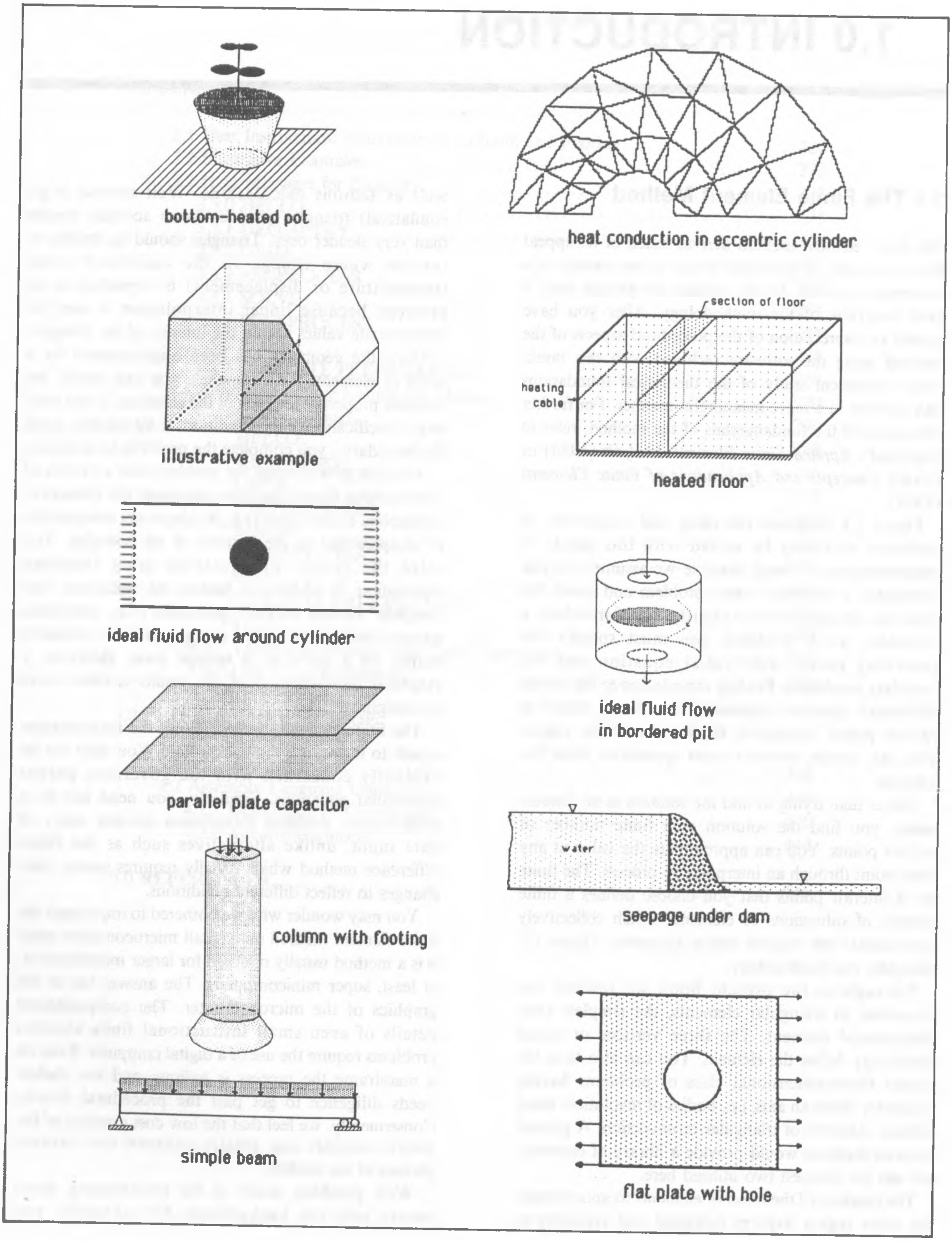
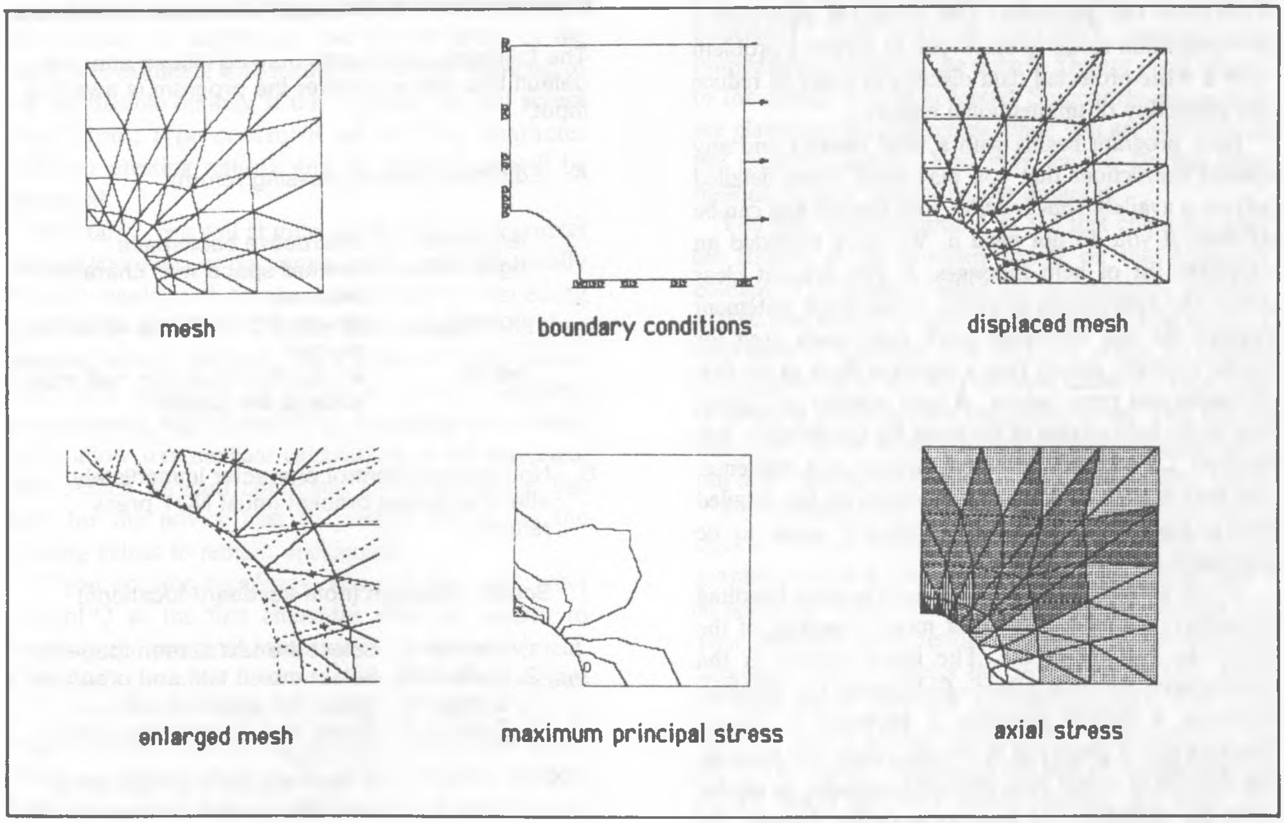


Fig. 1.2 Illustration of the finite element methodology



frequently have a special interest in values for only a few strategic locations but want to be assured that the solution at all points is 'well-behaved'. Graphics convey this information efficiently and effectively. In a similar manner, pictures facilitate comprehension of the data by describing the interconnection of the elements which form the global mesh.

Finally, the finite element method is an ideal candidate for a graphics-based computer implementation. We believe that you will be surprised by the ease with which you can move from heat conduction problems to elasticity problems even if you are not equally familiar with elasticity and heat conduction. We have structured these programs to make use of the commonality provided by the method.

1.2 The Apple II Implementation

We have concentrated our efforts on fully utilizing a 'plain vanilla' 64K Apple II Plus (with a RAM card installed), Apple II-e, or Apple II-c. Although you can use the program with only one disk drive, we think you will strongly prefer using two. No additional hardware is required. Even with these limited resources, you can successfully solve problems of the difficulty in the companion book by Segerlind. You can use the program for two-dimensional and axisymmetric finite element analysis for both steady state heat conduction and elasticity. Since the governing equation for heat conduction also describes diffusion, ideal fluid flow, and electrostatics, we have included examples of these as well.

After completing this introductory chapter, proceed to either the heat conduction or the elasticity chapter. Neither is a prerequisite for the other. However, the text and the programs reflect a common style. After studying one chapter, you will be able to master the other rather quickly.

We have tailored the programs for the novice user by the extensive use of menus that lead the user through the program. The main menu in each program lists the program options. Normally, you execute the parts sequentially. The default menu item number automatically advances. Simply press 'return' to accept the default. The data review items (list and plot) are optional. The exit option is given last and allows you to leave this program segment and advance to the next.

You can adjust the level of user interaction. At the maximal level of interaction, the program gives you numerous opportunities to display intermediate results and to monitor the computational details. When you become familiar with the details, you may wish to execute the program with minimal interaction after you formulate the problem. The program provides a demonstration mode to allow you to review a problem with a write-protected data diskette in order to reduce the possibility of unintentional erasure.

Each program begins with a brief abstract and any special instructions that you may need. More detailed advice is available but is under your control and can be avoided if you do not need it. We have provided an extensive list of help messages. If you are not clear about the appropriate response to an input statement (except the one keystroke GET commands used for cursor control), simply type a question mark as the first character and press 'return'. A help number will direct you to the help section of the book for clarification. See sections 2.3 and 3.3. Although written as a reference, you may wish to read these two sections for the detailed background information. Press 'return' again to be prompted for your input.

While we have provided an extensive error handling capability, we have been even more conscious of the desire to prevent errors. The input routine is the centerpiece for this effort. Following the prompt message, a default response is enclosed in square brackets and is displayed in inverse video. We have set the defaults to reflect your probable response, as we did with the indication of your progression through the principal menu of each program segment. We have provided default names for data files. If you retrieve a previously developed solution, in a demo mode for example, or make modifications such as mesh refinement, the retrieved values appear as the defaults whenever that is appropriate.

The visual verification afforded by the graphics support is probably the best aid for preventing or detecting input errors. An astonishingly large portion of the details in problem formulation diminish in importance when you have interactive graphics support. For example, when you have graphics, you typically need to know the numbering for only a few nodes. In contrast, without graphical support you must use the node numbers for the entire mesh and the opportunity for error increases dramatically.

Whenever any extended calculation is in progress, the program gives you reassuring evidence with messages, beeps, a sequence of dots, a moving '*' etc., that the calculations are proceeding. If the calculation fails or if you realize that your input is incorrect, various recovery techniques are available.

If you have not yet pressed the return key to accept input, you may use the arrow keys to edit your input. Control-X cancels the line; the left arrow erases individual keystrokes, while the right arrow recovers previously typed characters. The input command

Table 1.1 Input Commands

The following commands may be issued after the default brackets whenever the program is awaiting input.

A. Edit Input (before pressing return)

left arrow	destructive backspace
right arrow	forward space with character recovery
control-X	cancel the input and redisplay prompt
return	accept the input line; null input accepts the default

B. Non-printing control character immediately after the default bracket (must NOT press return)

Screen selection (note keyboard locations)

1. Control-T	Select the text screen (page 1)
2. Control-G	Select mixed text and graphics
3. Control-V	Select full graphics view
4. Control-P	Dump text or graphics screen to printer A user-supplied screen dump must be present to dump hi-res graphics. Text is also operational when the listing has been halted during a data review.

Program control

5. control-Q	Quit and return to principal menu
6. control-C	Cancel (exit) program

Program information

7. control-A	Display Applesoft pointers for memory usage
8. control-R	Display input restrictions

C. Single character immediately after the default bracket ('Return' is required.)

9. ?	Provide a help message number
10. <	Branch to previous input
11. >	Branch forward
12. :	Refer to a previously entered point (in GEOMETRY) only)

provides immediate checking for the maximum number of characters or keystrokes and for the range of the input. For example, with menus the input is restricted to the permissible options. If the program will not accept your input, type control-R as the first character (without pressing return); and the restrictions will be displayed.

Inevitably, you will at times enter valid but incorrect data. If you discover this immediately, you may usually branch backward to the immediately preceding question by entering '<' as the first character and then pressing 'return'. At times you will be sent back further than one question in order to avoid violating programming requirements (e.g., branching into a loop) or to allow intermediate calculations to be corrected. The branch forward command '>' is somewhat less easy for the novice user to predict but allows the existing values to remain unchanged.

If you become hopelessly confused, you may enter control-Q as the first character after the default to return to the principal menu for the program segment. Control-Q clears the stack of subroutine calls, executes a 'GOTO 2'(the restart address), and allows you to resume without erasing the variables. Should this fail to effect a recovery, two more drastic options remain. Either press control-C or reset (or control reset) to exit the program. Since the program saves more than twenty intermediate arrays, you may simply type 'RUN' and press 'return' to re-initialize the program variables and then re-load the intermediate array values. As an extreme measure you may turn the power off, reboot the program, and branch from the MAIN menu in HELLO to resume. Simply re-booting without 'power-down' does not assure that you have re-established the program and utilities in their entirety!

Because of the ease with which you may branch to any previous phase of the calculations, we have provided an enable/disable flag for each stored array. For example, if during the entry of the material properties you discover that the elements do not correctly match the variation in the properties, you may branch back to GEOMETRY to reformulate the problem. Then when you save the next recomputed array, all subsequent arrays, which are now incompatible, are flagged as unavailable but are not erased. (The demo node suppresses all data saves, except in PLOT, and allows you to intentionally and safely circumvent this data checking feature.) This disable feature does not apply to PLOT or to the data review options of listing and plotting in any of the program segment.

Only a few input options applicable throughout the programs remain to be described. You may toggle the video screen to text, to mixed graphics with four lines of text, or to full-view graphics at any time by entering control-T, control-G, or control-V, respectively, immediately following the default. Do not press 'return' or the default will be accepted. Sometimes instructions are placed on the text screen while you are working on the graphics screen. You may switch among these options without disrupting the program.

You may enter control-A as the first character after the default to get a status report on the Applesoft pointers. The memory usage by the program, variable, arrays, and strings is reported. Remember that memory allocation is dynamic. Not only can arrays be created during execution, but also we have added the capability of clearing arrays to free unneeded space. This feature not only allows more efficient memory management, but also enables the control-Q restart to be used freely.

For completeness, we again refer to the screen dump option. Control-P will print the currently visible text or graphics screen if you have a compatible printer and a hi-res dump utility. The text dump routine assumes that your interface card resides in slot 1. (The assembly language routine directs the output to $C102; some interface cards may use another address). You may need to add a printer initialization command in the HELLO program. Because of the lack of standardization in the hi-res graphics screen dumps, we have not provided a HI-RES DUMP program. If the user provides such a routine for the reserved RAM locations ($1D00.1FFF), the program will print the presently visible screen if control-P is pressed as the first character after the default brackets or while the output has been halted by the listing option which is invoked whenever tabular output is being listed. Table 1.1 summarizes these commands.

Table 1.2 illustrates the steps involved in solving a problem.

Program Organization

HELLO

You must activate this program first because it establishes the working environment for all the other program segments. DOS is moved to the RAM card; this HELLO program is loaded in memory above the hi-res graphics screen, and then the various utilities are loaded. If your computer supports lowercase characters, remember to set the caps lock. Next indicate the number of disk drives attached to your computer and specify the level of user interaction you desire. Finally select from the MAIN MENU the program segment you wish to execute next.

Table 1.2 Problem Solving Steps

1. HELLO

   Start-up and main program

2. GEOMETRY

   Create mesh generating curvilinear quadrilaterals

3. GRID

   Generate a mesh of triangular elements; modify mesh; automatically renumber nodes

4. PREPROCESS

   Input properties and boundary conditions

5. SOLVE

   Form equations and solve for nodal values

6. POSTPROCESS

   Use nodal values to compute additional quantities

7. PLOT

   Plot grid, boundary conditions, element and nodal values (contours, vectors, and shading)

8. DISKETTE PREPARATION

   Initialize data diskette

GEOMETRY

Geometry specifies the data required by the mesh generator in the next segment. In particular, you must define the global coordinate system and define coordinate points which will be used to specify the mesh generating regions. Each quadrilateral generating region requires eight such points and then must be traversed in a counterclockwise direction. Finally, specify whether you are solving a two-dimensional or an axisymmetrical problem and then specify the number of nodes of the elements you wish to place along each side of the curvilinear quadrilaterals.

You can use the full graphics support for problem formulation or you can enter the data without graphics, allowing a closer parallel with the input format of Segerlind. In addition, this nongraphics option permits you to create a more irregular mesh than can be done with the automatic mesh generator. Remember to pay careful attention to the counterclockwise convention in entering the data. Combining the graphics and nongraphics options can sometimes be an advantage. For example, you can generate a mesh with graphics and then modify the last region created. The program segment producing the tabular results may list the results. Some segments permit a graphical presentation of the results too.

GRID

GRID generates the mesh used to discretize the region. The program generates the element numbers, node numbers, and coordinates of the nodes and stores these on the diskette. The program determines the interconnections of the elements and provides you with an opportunity to modify the mesh by moving nodes or reversing diagonals. Next the program automatically renumbers the nodes to reduce the memory requirements for the problem and to reduce the computational time in subsequent program segments. Before you leave this program segment, the program automatically identifies the boundary lines and boundary nodes which are required when you specify the boundary conditions in the next segment.

PREPROCESS

With geometric characteristics (except thickness for two-dimensional problems) completely defined by the preceding segments, you can now specify the physical constants required to characterize the problem. You can complete the problem formulation by specifying the conditions to be maintained along each of the boundaries. See the TEXT program below (accessed through the PREPROCESS exit menu) for an outline of the use of this program as a pre-processor to formulate problems to be solved on a larger computer.

SOLVE

If the preceding steps have been completed correctly, you can find a unique solution for each of the remaining nodal unknowns - temperature for heat conduction and displacement components for elasticity. The solution process consists of the creation of a system of simultaneous linear algebraic equations, which incorporate the specified boundary conditions and which are then solved by Gaussian elimination.

POSTPROCESS

From the nodal values you can compute various other interesting characteristics. We are interested in quantities both at the nodes and at the element centroids.

For heat conduction these quantities include temperatures at the element centroids and temperature gradients both at the centroids and at the nodes. The equivalent nodal heat sources and resultant surface heat fluxes are also computed.

For elasticity these computed quantities of interest are the various strains and stresses at the nodes and centroids, the displaced positions of the nodes, and the nodal reaction forces.

PLOT

While the tabular computed results and a few plots were options in the previous program segments, we now concentrate on creating graphical output. You can generate plots of the discretizing mesh, the boundary conditions, and all the computed results. Contour plots are generated using linear interpolation on the elements. You can depict quantities with arrows which represent magnitude and direction, or you can use shading to represent various element quantities. A zoom capability provides a means to enhance image resolution. Using a single user-defined region, the program redraws the picture, clipping the unneeded parts. Alternatively, the program can subdivide the entire region into equal-sized zones which can be printed and then taped together to form a composite higher-resolution image. This process is rather time consuming but does permit higher quality results. The PLOT program does not produce tabular output.

DISKETTE PREPARATION

You must use this program to initialize a data diskette before the above steps can be executed. Because this program can erase the entire diskette destroying any existing files, care must be exercised in its use.

The DISKETTE PREPARATION program prepares the data diskette by adding a file directory of default file names. It can optionally initialize the diskette without an image of the DOS, thereby freeing extra space for data storage.

Tables 1.3 and 1.4 show the initial contents of these files. The two files differ in content, and you must not inadvertently swap a data diskette from heat to elasticity or vice-versa.

Table 1.3 FILEINFO.TXT Dump for Heat Conduction

Record #	Contents
(0)	1
	HEAT1
	HEAT.TEST1
(1)	0
	HEAT1/COORDINATES
	COORDINATES OF INPUT DATA POINTS
	1
	POINTS
	1
	DIRECTIONS
(2)	0
	HEAT1/REGIONS
	NODE AND CONNECTIVITY DATA
	1
	REGION(S)
	1
	NUMBERS EACH
(3)	0
	HEAT1/ELEMENT NODES
	ELEMENT NODE NUMBERS
	1
	ELEMENTS
	1
	NODES EACH
(4)	0
	HEAT1/NODE COORDS
	COORDINATES OF NODES
	1
	NODES
	1
	DIRECTIONS
(5)	0
	HEAT1/RENUM EL ND
	RENUMBERED ELEMENT NODE NUMBERS
	1
	ELEMENTS
	1
	NODES EACH
(6)	0
	HEAT1/RENUM ND COORDS
	COORDINATES OF RENUMBERED NODES
	1
	NODES
	1
	DIRECTIONS
(7)	0
	HEAT1/RENUM LINES
	RENUMBERED NODES FOR UNIQUE LINES
	1
	LINES
	1
	NODES/EL
(8)	0
	HEAT1/RENUM BOUNDS

RENUMBERED BOUNDARY NODES & ELEMENTS
1
NODES
1
NODES/EL
(9) 0
HEAT1/PROPERTIES
MATERIAL THERMAL PROPERTIES
1
ELEMENTS
1
PROPERTIES
(10) 0
HEAT1/INPUT BC
INPUT BOUNDARY CONDITION SPECIFICATIONS
1
INPUTS
1
PARAMETERS
(11) 0
HEAT1/BOUND COND
NODAL BOUNDARY CONDITIONS
1
NODES
1
CODE/VALUE
(12) 0
HEAT1/INIT SOURCES
INITIAL HEAT SOURCE VECTOR
1
NODES
1
VALUE
(13) 0
HEAT1/INIT STIFF
INITIAL GLOBAL STIFFNESS MATRIX
1
NODES
1
BANDWIDTH
(14) 0
HEAT1/CONV STIFFNESS
STIFFNESS INCLUDING CONVECTION B.C.
1
NODES
1
VALUE
(15) 0
HEAT1/MOD SOURCES
SOURCES MODIFIED BY TEMPERATURE B.C.
1
NODES
1
VALUE
(16) 0
HEAT1/MOD STIFF
STIFFNESS MODIFIED BY TEMPERATURE B.C.
1
NODES
1
BANDWIDTH
(17) 0
HEAT1/NODE TEMPS
NODAL TEMPERATURES
1
NODES
1
VALUE
(18) 0
HEAT1/ELEMENT TEMPS
AVERAGE ELEMENT TEMPERATURES
1
ELEMENTS
1
VALUES
(19) 0
HEAT1/EL TEMP GRADS
TEMPERATURE GRADIENTS AT ELEMENTS
1
ELEMENTS
1
COMPONENTS
(20) 0
HEAT1/ND TEMP GRADS
TEMPERATURE GRADIENTS AT NODES
1
NODES
1
COMPONENTS
(21) 0
HEAT1/ND HT SOURCES
NODAL EQUIVALENT HEAT SOURCES
1
NODES
1
VALUES
(22) 0
HEAT1/BOUNDARY FLUXES
HEAT FLUXES NORMAL TO BOUNDARIES
1
BOUNDS
1
NODES/VALUE

Record 0 contains (a) the element type code (1 = two-dimensional 3-node triangle and 2 = axisymmetric ring with 3-node triangular cross-section), (b) problem keyword used to identify a set of related files, and (c) a problem description of up to 80 characters. Colons and commas are excluded.

Records 1-24 contain a status indicator (0 = inactive; 1 = active), the data file name, the data file descriptor, the maximum row subscript, the array row-descriptor, the maximum column subscript, and the array column-descriptor.

Table 1.4 FILEINFO.TXT Dump for Elasticity

Record #	Contents
(0)	1
	ELAST1
	TEST.ELAST
(1)	0
	ELAST1/COORDINATES
	COORDINATES OF INPUT DATA POINTS
	1
	POINTS
	1
	DIRECTIONS
(2)	0
	ELAST1/REGIONS
	NODE AND CONNECTIVITY DATA
	1
	REGION(S)
	1
	NUMBERS EACH
(3)	0
	ELAST1/ELEMENT NODES
	ELEMENT NODE NUMBERS
	1
	ELEMENTS
	1
	NODES EACH
(4)	0
	ELAST1/NODE COORDS
	COORDINATES OF NODES
	1
	NODES
	1
	DIRECTIONS
(5)	0
	ELAST1/RENUM EL ND
	RENUMBERED ELEMENT NODE NUMBERS
	1
	ELEMENTS
	1
	NODES EACH
(6)	0
	ELAST1/RENUM ND COORDS
	COORDINATES OF RENUMBERED NODES
	1
	NODES
	1
	DIRECTIONS
(7)	0
	ELAST1/RENUM LINES
	RENUMBERED NODES FOR UNIQUE LINES
	1
	LINES
	1
	NODES/EL
(8)	0
	ELAST1/RENUM BOUNDS
	RENUMBERED BOUNDARY NODES & ELEMENTS
	1
	NODES
	1
	NODES/EL
(9)	0
	ELAST1/PROPERTIES
	ELASTIC & THERMAL PROPERTIES
	1
	ELEMENTS
	1
	PROPERTIES
(10)	0
	ELAST1/INPUT BC
	INPUT BOUNDARY CONDITION SPECIFICATIONS
	1
	INPUTS
	1
	PARAMETERS
(11)	0
	ELAST1/BOUND COND
	NODAL D.O.F. BOUNDARY CONDITIONS
	1
	D.O.F.
	1
	CODE/VALUE
(12)	0
	ELAST1/INIT FORCES
	INITIAL (THERMAL) FORCE VECTOR
	1
	D.O.F.
	1
	VALUE
(13)	0
	ELAST1/INIT STIFF
	INITIAL GLOBAL STIFFNESS MATRIX
	1
	D.O.F.
	1
	BANDWIDTH
(14)	0
	ELAST1/COMB FORCES
	COMBINED INITIAL AND BOUNDARY FORCES
	1
	D.O.F.
	1
	VALUE
(15)	0
	ELAST1/MOD FORCES
	FORCES MODIFIED BY DISPLACEMENT B.C.
	1
	D.O.F.
	1
	VALUE
(16)	0
	ELAST1/MOD STIFF
	STIFFNESS MODIFIED BY DISPLACEMENT B.C.
	1
	D.O.F.

1
BANDWIDTH
(17) 0
ELAST1/NODE DISP
NODAL DISPLACEMENT MAGNITUDES
1
D.O.F.
1
VALUE
(18) 0
ELAST1/EL STRAINS
ELEMENT STRAINS IN COORD
DIRECTIONS
1
ELEMENTS
1
COMPONENTS
(19) 0
ELAST1/EL STRESSES
ELEMENT STRESSES IN COORD
DIRECTIONS
1
ELEMENTS
1
COMPONENTS
(20) 0
ELAST1/EL P STRESS
ELEMENT PRINCIPAL STRESSES &
DIRECTIONS
1
ELEMENTS
1
MAGN/DIR
(21) 0
ELAST1/NODE STRESS
NODAL STRESSES IN COORD
DIRECTIONS
1
NODES
1
COMPONENTS
(22) 0
ELAST1/ND PR STRESS
NODAL PRINCIPAL STRESSES &
DIRECTIONS
1
NODES
1
MAGN & DIR
(23) 0
ELAST1/NEW COORDS
NEW COORDINATES OF NODES
1
NODES
1
DIRECTIONS
(24) 0
ELAST1/ND REACTIONS
RESULTING REACTIONS AT NODES
1

TEXT

TEXT is an outline of a program which will permit you to use this microcomputer implementation as a pre-processor for data input for a larger computer. You can produce either a printed or a diskette version of output to facilitate the keyboard or modem transfer process. To formulate substantially larger problems, you will probably need to limit the generality of the material properties array to a smaller set of coded values, as is done by Gustafson (1977).

After this somewhat general introduction, you are ready to proceed to either the Heat Conduction (Chapter 2) or Elasticity (Chapter 3) Program.

2.0 STEADY STATE HEAT CONDUCTION,
Ideal Fluid Flow, Electrostatics, Seepage and Other Processes Governed by Poisson's Equations

This chapter deals with steady state heat conduction, ideal fluid flow, electrostatics, seepage, and other processes governed by Poisson's equation as well as the special case, Laplace's equation. The numerous examples provided in this chapter illustrate the generality of this finite element program. While the program handles this general class of boundary value problems, we elected to use the language of steady state heat conduction to reduce the level of abstraction required of the novice. See Segerlind (1984, 87-89) and Moon and Spencer (1961) for illustrations.

2.1 User Instructions: A Bottom-Heated Flower Pot

For brevity and clarity, we use the following step-by-step instructions for a specific example to illustrate the general process.

We assume that you have an Apple II (Plus, -e or -c) computer with a monochrome monitor and two disk drives. If you have only one drive, you must follow the diskette swapping instructions carefully. You need only a minimal familiarity with the operating characteristics of the Apple II to follow these instructions.

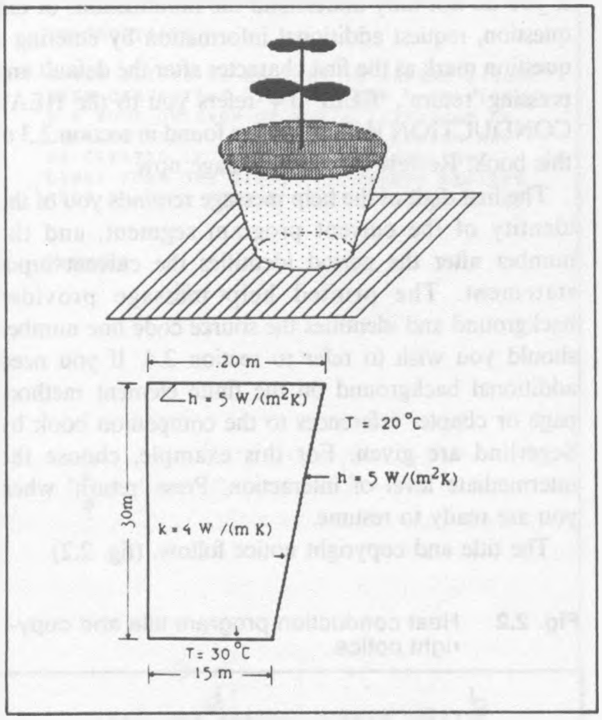
First, make a working copy of the unprotected Heat Conduction Program Diskette, store the original in a safe place, and write protect the working copy. In addition to the working copy of the program diskette, you need a blank data diskette. Since diskettes are fragile, protect them from dust and fingerprints.

Here is a statement of the problem.

To reduce greenhouse heating costs and to promote faster and more uniform plant growth, we wish to explore the use of direct heating of the plant through its roots rather than through its leaves. Suppose we heat the bottom of the nursery container directly (see fig. 2.1), rather than elevating the air temperature in the poorly insulated greenhouse. The design of such a system requires a knowledge of the temperature throughout the potting material. (See Yang and Albright 1985 for additional details.)

Figure 2.1 also shows the cross-section of a tapered, axisymmetric pot in a 20 C room placed on a heated 30 C bench. Find the internal temperature profile and temperature gradients.

Fig. 2.1 Bottom-heated flower pot (Yang and Albright 1985)



As an important first step, make a sketch of the problem (fig. 2.1b) to guide your analysis. Since the bottom surface has the highest temperature and the room has the lowest, the top right corner of the pot should be its coolest point. However, the largest temperature gradients should be at the bottom right where the change from 30 to 20 occurs over the shortest distance. This suggests that the mesh you generate should be more refined in that vicinity in order to improve the accuracy of the solution.

Since you can consider the thermal conductivity to be uniform, you only need a single quadrilateral mesh generating region. The four vertices correspond to the four corners of the cross-section as shown in Figure 2.1b. The placement of the remaining four 'mid-side' points affects the local refinement of the generated mesh. A midpoint placement results in equal-sized intervals. Move the bottom and right side points closer to the bottom right corner to make the elements smaller in that corner.

To begin the program, place the program diskette in drive 1 and turn on the monitor and computer. After several startup messages, you must specify the number of disk drives you are using. If you have two, press 'return' to accept the default shown in reverse video enclosed in square brackets. If you have only one drive, you must have the program diskette in the drive when you change program segments; but the data diskette must be resident at all other times with one exception in PLOT. Next specify the desired level of user interaction. If you do not fully understand the ramifications of this question, request additional information by entering a question mark as the first character after the default and pressing 'return'. 'HELP 0.4' refers you to the HEAT CONDUCTION Help Messages found in section 2.3 of this book. Review the help message now.

The first digit of the help message reminds you of the identity of the current program segment, and the number after the period identifies the current input statement. The printed help message provides background and identifies the source code line number should you wish to refer to section 2.4. If you need additional background on the finite element method, page or chapter references to the companion book by Segerlind are given. For this example, choose the intermediate level of interaction. Press 'return' when you are ready to resume.

The title and copyright notice follow. (fig. 2.2)

Fig. 2.2 Heat conduction program title and copyright notice

```
APPLIED FINITE ELEMENT ANALYSIS:
  AN APPLE ][ IMPLEMENTATION

      HEAT CONDUCTION

            BY

J. R. COOKE AND D. C. DAVIS

      COPYRIGHT, 1985

  JOHN WILEY AND SONS, INC.

    ALL RIGHTS RESERVED
```

Press 'return' again to reach the most important menu (fig. 2.3) in the entire program, referred to throughout as the MAIN MENU.

Fig. 2.3 The MAIN MENU

```
***   MAIN PROGRAM MENU (0.00)       ***

1. GEOMETRY
   DEFINE PROBLEM GEOMETRY
2. GRID
   GENERATE & MODIFY GRID
3. PREPROCESS
   DEFINE PROPERTIES AND BOUNDARY
   CONDITIONS
4. SOLVE
   ASSEMBLE EQUATIONS & SOLVE FOR NODE
   TEMPERATURES
5. POSTPROCESS
   SOLVE FOR AVERAGE TEMPERATURES AND
   GRADIENTS
6. PLOT
   PLOT GRID, BOUNDARY CONDITIONS, &
   PROBLEM RESULTS
7. DISKETTE PREPARATION
   PREPARE DATA DISKETTE

0. STOP
```

Table 1.2 outlines the overall program organization and the function of each of the program segments. To solve the problem proceed sequentially through the first six items. The default prompt keeps track of your progress. If you skip one of the items, certain intermediate results needed to complete a subsequent step will not be generated. On the other hand, if you return to an earlier program segment, the program disables all data files dependent upon the revised file you create. This safety feature ensures that your data files are internally compatible.

DISKETTE PREPARATION

Since this is your first exercise, you must prepare a data diskette before you can proceed with the problem formulation. We made DISKETTE PREPARATION the last executable item in the menu to reduce the possibility of accidental diskette erasure. Select option 7 and press 'return'. Insert the data diskette as instructed. Be sure that you write-protect the working copy of the program diskette by putting a tab on the diskette, since a wrong move here could be disappointing. Press 'return' when ready.

This is your last opportunity to review the current contents of the data diskette before you must decide whether to reformat. As a safety step, you must authorize diskette erasure with 'Y' and 'return'. The diskette produced has 543 free sectors, 47 more than you would have gotten with the usual INIT command.

Provide an identifying descriptive keyword of up to 8 characters, as indicated by the number of underscores. Choose HEAT1 to remind you of 'bottom-heated pot example 1'. This keyword will be the prefix for each of the related data files. Commas and colons are prohibited here and in the more complete description which you enter next. Press 'return' and wait for the directory 'FILEINFO.TXT' to be created. Table 1.3 describes the file contents. Return to the MAIN MENU.

Problem formulation always begins at the MAIN MENU to assure the proper configuration of memory. Re-select the intermediate level of user interaction. This intentionally repeated question allows you to return to the MAIN MENU at any point and change this option.

Select GEOMETRY and press 'return' and the program segment will be loaded and activated.

GEOMETRY

GEOMETRY begins with an abstract (fig. 2.4).

Fig. 2.4 GEOMETRY abstract

```
        APPLIED FINITE ELEMENT ANALYSIS:
          AN APPLE ][ IMPLEMENTATION

               COPYRIGHT, 1985
               GEOMETRY  (1.00)

  ABSTRACT:

  DISCRETIZATION BEGINS WITH THE CREATION
  OF CURVILINEAR QUADRILATERAL SUB-REGIONS
  WHICH TOGETHER DESCRIBE THE OVERALL
  GEOMETRY.

  SUBSEQUENTLY, THESE INPUT REGIONS WILL
  BE SUB-DIVIDED INTO A MESH OF 3-NODE
  TRIANGLES.
```

The program library contains only two element types. Select the axisymmetric 3-node triangle for this problem.

Each program segment has a principal menu numbered (X.05) to guide your progress (fig. 2.5). Select 'input with graphics'. Since you are not revising or reviewing an existing data file, you cannot retrieve the data. If such a data file did exist and were retrieved, the defaults would reflect this data.

Fig. 2.5 GEOMETRY principal menu

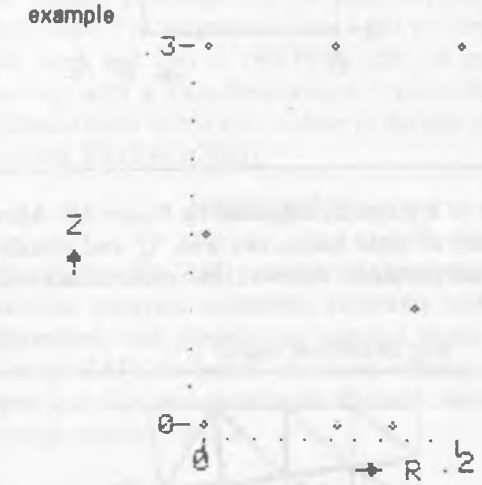
```
***        GEOMETRY (1.05)        ***

 --- POINT COORDINATE ENTRY ---
1. INPUT WITH GRAPHICS
2. INPUT WITHOUT GRAPHICS

  ---  REGION DEFINITION/DIVISION  ---
3. INPUT WITH GRAPHICS
4. INPUT WITHOUT GRAPHICS

  ---  DATA REVIEW   ---
5. LIST INPUT DATA
6. PLOT INPUT DATA

0. NONE OF THE ABOVE
```

Enter the data shown in figure 2.1. As indicated in figure 2.6, establish the global coordinate system with 0 <= R <= 0.20m and 0 <= Z <= 0.30m. If you enter a value incorrectly, use the arrow keys to edit the input before you press 'return'. If you do not discover an input error until after pressing 'return' use '<' as the first input character after the default and press 'return' to go back to the previous question. If you wish to abandon your current formulation, enter control-Q as the first character without pressing 'return'; and you will be returned to the principal menu to restart this step.

Fig. 2.6 Instructions for input of coordinates

```
GEOMETRY: COORDINATES (1.31)

INSTRUCTIONS:

ENTER POINTS TO BE USED TO DEFINE 8-NODE
MESH GENERATING REGIONS WITH 4 VERTICES
& A NODE ON EACH OF THE 4 (STRAIGHT OR
CURVED) SIDES. A TRIANGULAR REGION MAY
BE CREATED IF TWO CONNECTED STRAIGHT
LINES FORM ONE SIDE.   ADJACENT REGIONS
MUST SHARE 3 POINTS ON THE COMMON SIDE.
```

example



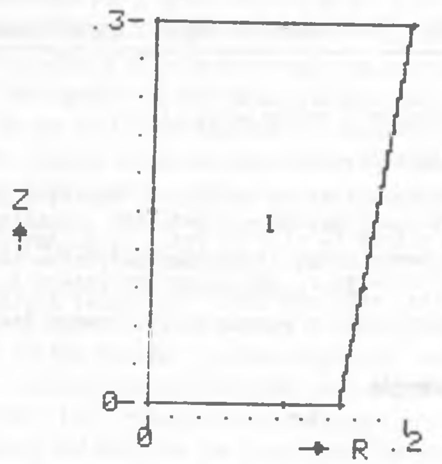
Enter (E) the 8 points needed to define the mesh generating region shown in figure 2.7. Enter the points individually or with straight line segments. You can generate the mid-side points automatically using the Line (L) option. To generate smaller triangles at the bottom right corner, place the 'mid-side' points on the bottom and right sides closer to that corner as shown in figure 2.1. A fractional distance along the directed line is suggested on figure 2.1. Select 'Q' to leave the input of points.

From the GEOMETRY menu proceed to region definition with graphics. Beginning at a vertex, create region 1 by defining a COUNTERCLOCKWISE

Fig. 2.7 Instructions for region definition

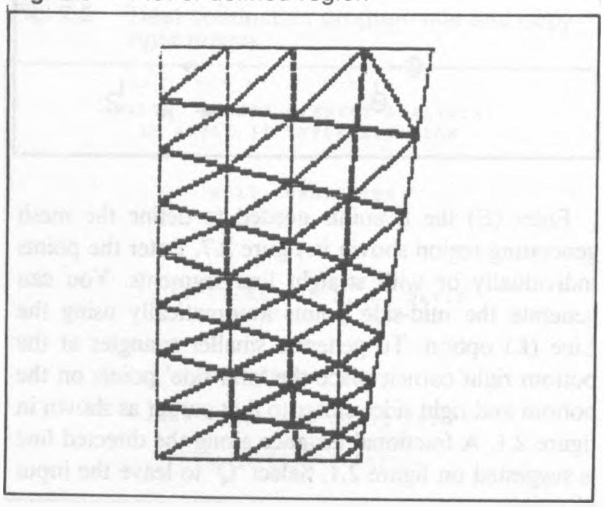
```
GEOMETRY: REGION DEFN'S (1.32)

INSTRUCTIONS:

BEGINNING WITH A VERTEX, DEFINE EACH
REGION WITH A COUNTER-CLOCKWISE SEQUENCE
OF 8 POINTS.

USE ARROWS TO POSITION THE CURSOR. THEN
PRESS <S> TO SELECT.

EACH REGION IS NUMBERED AT ITS CENTROID.
```

example



sequence of 8 points as suggested by Figure 2.8. After you specify all eight points, exit with 'Q' and visually verify that you have followed the counterclockwise rule.

Fig. 2.8 Plot of defined region

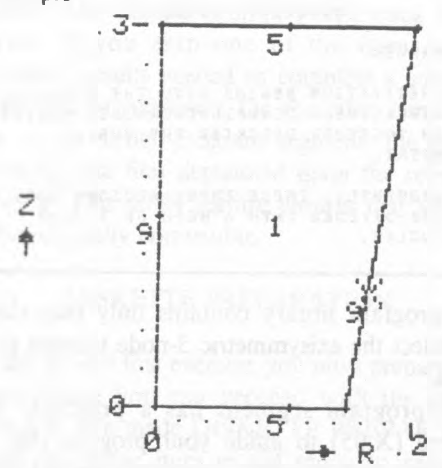


The program automatically identifies common region boundaries. For more complex problems, you will need multiple regions having common sides. To complete the creation of the mesh (fig. 2.9), designate the number of element nodes to be placed along each side of the mesh generating region. Point to a side by moving the 'X' cursor to it and select it by pressing 'S'. Enter the number of nodes per side as shown in Figure 2.9.

Fig. 2.9 Instructions and plot of nodes per side

```
GEOMETRY: NODES PER SIDE  (1.33)

INSTRUCTIONS:
   THE NUMBER OF TRIANGULAR ELEMENT
NODES ON EACH CURVILINEAR QUADRILATERAL
REGION SIDE (INCLUDING BOTH END NODES)
IS >=2. OPPOSITE REGION SIDES MUST HAVE
THE SAME NUMBER OF NODES; THIS IS AUTO-
MATICALLY ENFORCED.

USE ARROWS TO POSITION THE CURSOR. THEN
PRESS <S> TO SELECT.

ENTER THE NUMBER OF NODES FOR THE MARKED
SIDES. CHANGE ENTRIES BY RE-SELECTING
THE SIDE.
```

example



When you generate a mesh, remember that the elements in a mesh are connected to other elements only through shared nodes (i.e., vertices of the triangles). Consequently, when you work with multiple mesh generating quadrilaterals, the common boundaries must have the same number of element nodes. The program automatically assures this condition if the automatic mesh generator is used.

Review the data generated in this program segment (options 5 and 6) before selecting option 0 to proceed to GRID generation.

GRID

The abstract reminds you of the purpose of this program segment (fig. 2.10). You can execute this

Fig. 2.10 GRID abstract

```
    APPLIED FINITE ELEMENT ANALYSIS:
       AN APPLE ][ IMPLEMENTATION


            COPYRIGHT, 1985

              GRID  (2.00)


 ABSTRACT:

 A GRID OF  3-NODE TRIANGULAR ELEMENTS IS
 GENERATED FOR THE PROBLEM DOMAIN.  NEXT,
 NODES ARE RENUMBERED  TO REDUCE  PROBLEM
 BANDWIDTH.  THEN UNIQUE LINES AND  NODES
 ON BOUNDARIES ARE IDENTIFIED.
```

program segment in a local free-run mode using option 1 (fig. 2.11), or you can be more deliberate and retain your ability to examine internal details by sequentially performing options 2-4 individually. On this first example, select option 1 and run the program segment without user interaction. The program retrieves data from GEOMETRY and calculates the nodal coordinates.

Fig. 2.11 GRID principal menu

```
   ** GRID   (2.05) **

  -- GRID DEFINITION --
 1. DO COMPLETE GRID GENERATION (2-4)

 2. DEFINE GRID ELEMENTS & NODES
 3. RENUMBER NODES TO REDUCE BANDWIDTH
 4. SORT UNIQUE LINES & BOUNDARY NODES

  -- GRID EXAMINATION --
 5. LIST ELEMENT NODAL COORDINATES
 6. LIST LINE & BOUNDARY NODES
 7. PLOT ENTIRE GRID

 0.NONE OF THE ABOVE
```

The mesh for each region is plotted separately. Since only one region is required here, a composite mesh is not drawn. The mesh has 56 elements and 40 nodes. The 'bandwidth' of 7 stipulates that the maximum difference in related node numbers is 6 (i.e., one less than the number given). (See Segerlind 1984, 47-50 for a more detailed discussion.) At this point you need only recognize that this number is related to the width of the largest array you will generate, and, hence, provides a clue to the degree of mesh refinement this program can handle. The current formulation certainly does not tax memory limitations.

Here, or at any input request, examine memory usage by entering control-A immediately after the default bracket (without pressing 'return'). Note that more than 9K of RAM is unused at this point. Typically, RAM does not limit this step. 'Related nodes' (i.e., those nodes connected to the node in question by a single element side) are needed for the bandwidth reduction step. The program uses the Collins algorithm to generate nodal renumberings that might reduce the bandwidth and, hence, reduce RAM requirements and computation time in subsequent program segments. The algorithm does not guarantee an optimal numbering; you retain the original numbering if no reduction is achieved.

The program identifies unique lines and the boundary nodes. You can list the data for the generated grid using options 5 and 6. Press 'return' for each additional screen of data or press the space bar to advance by single lines. Use control-P to print this output. You can use this graphically generated data as input for Segerlind's non-graphics batch programs. Use option 7 to visually review the mesh. Review the node numbering by using the left and right arrows to move the cursor. You need identify only selected nodes to use with the results yet to be calculated. You can relegate the numbering of the interior points to a secondary role because of the graphical support.

Now you have completed the geometrical formulation and can proceed to PREPROCESS to the input of the thermal properties and the boundary conditions. Select item 0 of the principal menu and proceed to the exit menu and then to PREPROCESS. (If you were dealing with a two-dimensional formulation, the thickness input would also be done in the next program segment, PREPROCESS).

PREPROCESS

The program (fig. 2.12) retrieves data generated in the previous program segments, generates additional information, and presents the principal menu (3.05). (See fig. 2.13). As before, the menu consists of data input and data review sections. Proceed sequentially through options 1 and 2.

Fig. 2.12 PREPROCESS abstract

```
    APPLIED FINITE ELEMENT ANALYSIS:
       AN APPLE ][ IMPLEMENTATION


            COPYRIGHT, 1985

          PREPROCESS (3.00)

 ABSTRACT:

 DEFINE MATERIAL PROPERTIES AND BOUNDARY
 CONDITIONS FOR THE PROBLEM.  MATERIAL
 PROPERTIES MUST BE DEFINED  BEFORE B. C.


 MATERIAL PROPERTIES INCLUDE THERMAL
 CONDUCTIVITIES AND ELEMENT THICKNESS
 (FOR 2-DIMENSIONAL PROBLEMS).  BOUNDARY
 CONDITION  OPTIONS  ARE:  HEAT  SOURCES,
 SPECIFIED  TEMPERATURES,   HEAT  FLUXES,
 AND  CONVECTION.
```

Fig. 2.13 PREPROCESS principal menu

```
*** PROPERTIES & BOUND. COND. (3.05) ***

 --- DATA INPUT ---
1. INPUT MATERIAL THERMAL PROPERTIES AND
   THICKNESS
2. INPUT TEMPERATURE, FLUX, CONVECTION &
   SOURCE BOUNDARY CONDITIONS

 --- DATA EXAMINATION ---
3. LIST PROPERTIES
4. PLOT PROPERTIES
5. LIST INPUT BOUNDARY CONDITIONS
6. LIST EQUIV. BOUNDARY CONDITIONS
7. PLOT INPUT BOUNDARY CONDITIONS
8. PLOT EQUIV. BOUNDARY CONDITIONS

0. NONE OF THE ABOVE
```

Assume that the thermal conductivity is not directional; therefore, you are dealing with an isotropic problem. Furthermore, assume the property is constant everywhere and choose to enter the single value [4W/(mK)] for the entire body (fig. 2.1).

Although you can specify the boundaries at individual nodes or for a continuous group of nodes, use the latter. Along the bottom a constant temperature condition (30 C) is prescribed. On the tapered side and the top you know the temperature of the surrounding fluid (20 C) and the convection coefficient. Because of symmetry no heat is conducted across the left side of the figure. A zero gradient is the automatic default assumed by the program for a boundary; therefore, you need not enter this condition. Use 'Q' to quit. The program computes the nodal equivalent of the boundary conditions before returning you to the principal menu.

Review the properties and boundary conditions using options 3-8. The input specifications for a convection boundary condition apply to the side of an element (i.e., between two nodes). Note the handling of the nodes when a change occurs in the boundary condition. Finally, note that the 'Boundary (out) convection' value corresponds to the product of the conductance coefficient and the fluid temperature. (See Segerlind's equation 9.2 and pages 145-148). Menu Option 6 shows that the nodes along the axis of symmetry have actually been prescribed as a no external flow boundary.

If the thermal properties had not been uniform throughout, the properties could have been specified on individual element or individual region bases. You can assign letter codes for up to six different values to expedite input. Proceed to SOLVE.

SOLVE

You have completed the input and the program can now form the system of linear algebraic equations. As indicated by the abstract (fig. 2.14), you define and then assemble (by superposition) the equations for each element to form the global equations. Then these global equations are modified to assure that the solution is consistent with the nodal equivalent boundary conditions. Gaussian elimination is used to solve for the nodal temperatures (Review Segerlind 1984, 47-50.)

Fig. 2.14 SOLVE abstract

```
  APPLIED FINITE ELEMENT ANALYSIS:
     AN APPLE ][ IMPLEMENTATION

          COPYRIGHT, 1985

           SOLVE (4.00)
ABSTRACT:

FINITE ELEMENT EQUATIONS ARE DEFINED FOR
EACH ELEMENT, ASSEMBLED INTO A GLOBAL
MATRIX EQUATION, MODIFIED  BY BOUNDARY
CONDITIONS, AND SOLVED FOR NODE
TEMPERATURES.
```

The principal menu (4.05) allows you to execute options 2-5 without interruption (fig. 2.15). If you execute the options individually to display more detail, you must preserve the sequence of the steps. Use option 6 to review the nodal temperatures before proceeding to POSTPROCESS.

Fig. 2.15 SOLVE principal menu

```
*** PROBLEM SOLUTION (4.05) ***

 --- FORM & SOLVE ALL EQUATIONS ---
1. DO COMPLETE PROBLEM SOLUTION (MENU
   OPTIONS 2 - 5)

 --- FORM SYSTEM EQUATIONS ---
2. ASSEMBLE GLOBAL FORCE (SOURCE) AND
   STIFFNESS (COEFFICIENT) MATRICES
3. APPLY CONVECTION BOUNDARY CONDITIONS
4. APPLY CONSTANT TEMPERATURE BOUNDARY
   CONDITIONS

 --- SOLVE SYSTEM EQUATIONS ---
5. SOLVE FOR NODE TEMPERATURES

 --- LIST RESULTS ---
6. LIST NODE TEMPERATURES

0. NONE OF ABOVE
```

If you only need a plot of the nodal temperatures, you may proceed immediately from the principal menu (5.05) to PLOT. Frequently, however, you need to find various temperature gradients and heat fluxes.

POSTPROCESS

In POSTPROCESS (fig. 2.16 and 2.17) the principal menu (5.05) indicates that this program segment, like

SOLVE, is computation bound and requires no additional user-input. Use option 1 to find the element temperature, temperature gradient, and nodal temperature gradients. Similarly, compute the resultant nodal heat sources and surface heat fluxes. You can ONLY list each of these classes of computed results here. You will ONLY be able to produce plots in the final program segment PLOT.

Fig. 2.16 POSTPROCESS abstract

```
        APPLIED FINITE ELEMENT ANALYSIS:
           AN APPLE ][ IMPLEMENTATION

                COPYRIGHT, 1985

               POSTPROCESS (5.0)

   ABSTRACT:

   NODE TEMPERATURES DETERMINED IN 'SOLVE'
   ARE USED TO DEFINE ELEMENT TEMPERATURES,
   TEMPERATURE GRADIENTS, NODE HEAT SOURCES
   AND HEAT FLUXES.  ANY OR ALL OF THESE
   MAY BE DETERMINED.
```

Fig. 2.17 POSTPROCESS principal menu

```
          * POST-PROCESSING (5.05) *
       - SOLVE ALL EQUATIONS -
    1. DO ALL CALCULATIONS (MENU OPTIONS
       2 - 6)

       - SOLVE FOR TEMPERATURES & GRADIENTS -
    2. ELEMENT TEMPERATURES
    3. ELEMENT TEMPERATURE GRADIENTS
    4. NODE TEMPERATURE GRADIENTS

       - SOLVE FOR SOURCES & FLUXES -
    5. RESULTANT NODE HEAT SOURCES
    6. RESULTANT SURFACE HEAT FLUXES

       - LISTING -
    7. LIST OUTPUT

    0. NONE OF THESE
```

PLOT

PLOT produces plots of the mesh, the boundary conditions, and the computed results. (See fig. 2.18 and 2.19.) To increase resolution you may draw enlargements. You can either enlarge an arbitrary user-specified rectangular portion or produce several multiple plots using a common enlargement in order to produce a large composite picture. An elaborate labelling capability allows you to produce finished plots.

Using the conventions followed throughout the program, you may exercise considerable control over label placement. Use the left and right arrows to move the cursor. Press 'S' to select a label to be changed. An

Fig. 2.18 PLOT abstract

```
        APPLIED FINITE ELEMENT ANALYSIS:
           AN APPLE ][ IMPLEMENTATION

                COPYRIGHT, 1985

                  PLOT  (6.0)

   ABSTRACT:

   PREPARE FINISHED PLOTS OF GRID, BOUNDARY
   CONDITIONS, OR RESULTS.  ANY PLOT MAY BE
   LABELED AND PRINTED.  ZONE PLOTS  MAY BE
   SELECTED TO ENLARGE PARTS OF INTEREST.
```

Fig. 2.19 PLOT principal menu

```
              PLOT  (6.05)

          -- PLOT FORMULATION --
       1. PLOT GENERATED GRID
       2. PLOT BOUNDARY CONDITIONS

          -- PLOT RESULTS --
       3. PLOT NODE TEMPERATURES
       4. PLOT ELEMENT TEMPERATURES
       5. PLOT NODE TEMP GRADIENTS
       6. PLOT ELEMENT TEMP GRADIENTS

          -- OTHER --
       7. RETRIEVE PICTURE FROM DISK

       0. NONE
```

'X' adds or removes the label. After a label has been selected by 'S' use the traditional I, J, K, and M diamond to move the label in large increments. Smaller increments of movement are possible using control-I, control-J, control-K, and control-M. Use 'Q' to quit the modification of the selected label. Two successive 'Q''s allow you to exit the label edit routine. A 're-draw' option allows you to erase the label background and destructively (DRAW rather than XDRAW) re-write the label.

Use the principal menu options 1-6 (fig. 2.19) in any order to display the previously generated data files. (If you are solving a very large problem, you may need to eliminate unnecessary arrays by doing a cold restart (exit and RUN) and then proceeding directly to a specific plot.) Figure 2.20 illustrates the various types of plots. Use option 7 to retrieve an existing picture file saved earlier in options 1-6.

As an exercise to test your mastery of the program, solve the related problem shown in figure 2.21. The soil temperature varies substantially, and the thin plastic potting containers provide very little insulation. Suppose a layer of insulation were added to the side walls. Does this improve the performance of this technique? Hint: Assume perfect insulation along the outer sloping surface and set the normal gradient to zero.

Fig. 2.20 Mesh, boundary conditions, and constant temperature contours for bottom-heated flower pot

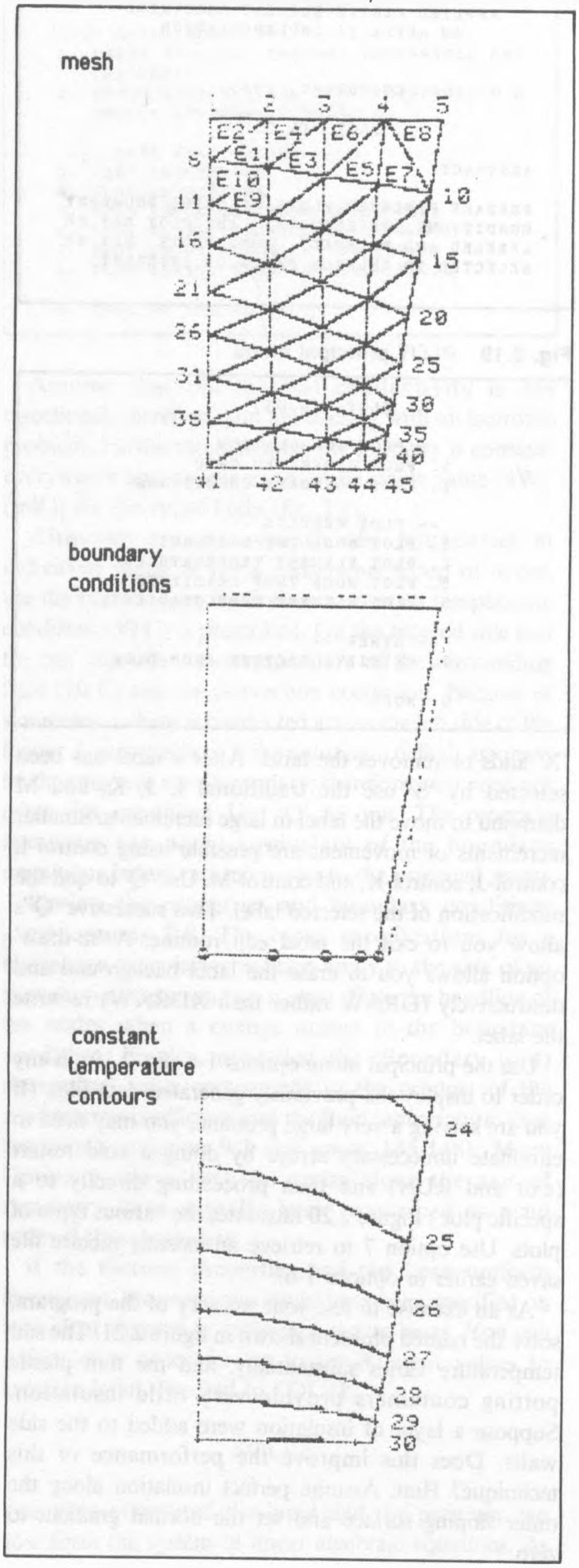
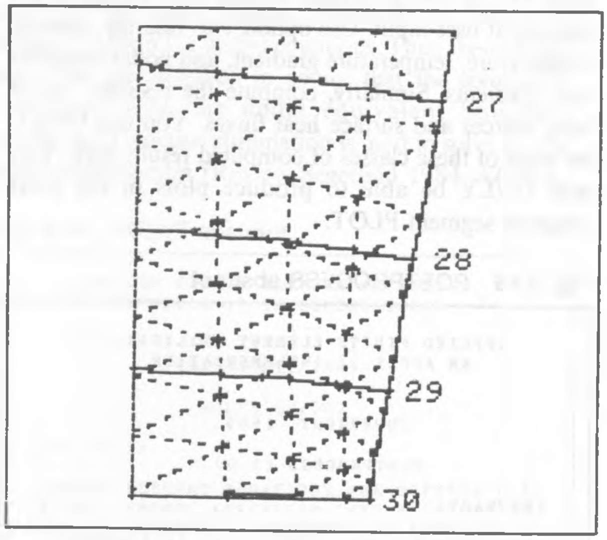


Fig. 2.21 Isotherms for a perfectly insulated pot



2.2 Illustrative Examples

We include the following collection of examples to illustrate the range of problems which you can solve using the 'Heat Conduction' program. In addition, these examples illustrate automatic mesh generation techniques.

While we present only four categories of problems, the actual class of Laplace equation governed processes is much broader. We include classical problems and contemporary problems as well as references for additional background and solution verification. Other references are included as a source of additional exercises.

2.2.0 Heat Conduction

Steady state heat conduction has a broad appeal and includes a wealth of interesting applications of Laplace's equation. Many individuals intuitively grasp heat conduction more easily than other mathematically equivalent areas; therefore, we present this area first.

Bottom-heated Insulated Flower Pot

Return to the bottom-heated flow pot example in the previous section and explore a refinement. You wish to maintain a suitable soil temperature throughout the pot. Since the root system will fill the entire pot, you wish to increase the temperature near the perimeter and especially near the top of the pot (fig. 2.22). Supply a layer of insulation around the outside of the pot - excluding the bottom and top. Figure 2.23 sketches the solution. You might make a limiting check by solving the problem with the normal gradient set to zero on the outer sloping surface if you did not complete this exercise in Section 2.1.

Fig. 2.22 Bottom-heated pot with insulated side (mesh and boundary conditions)

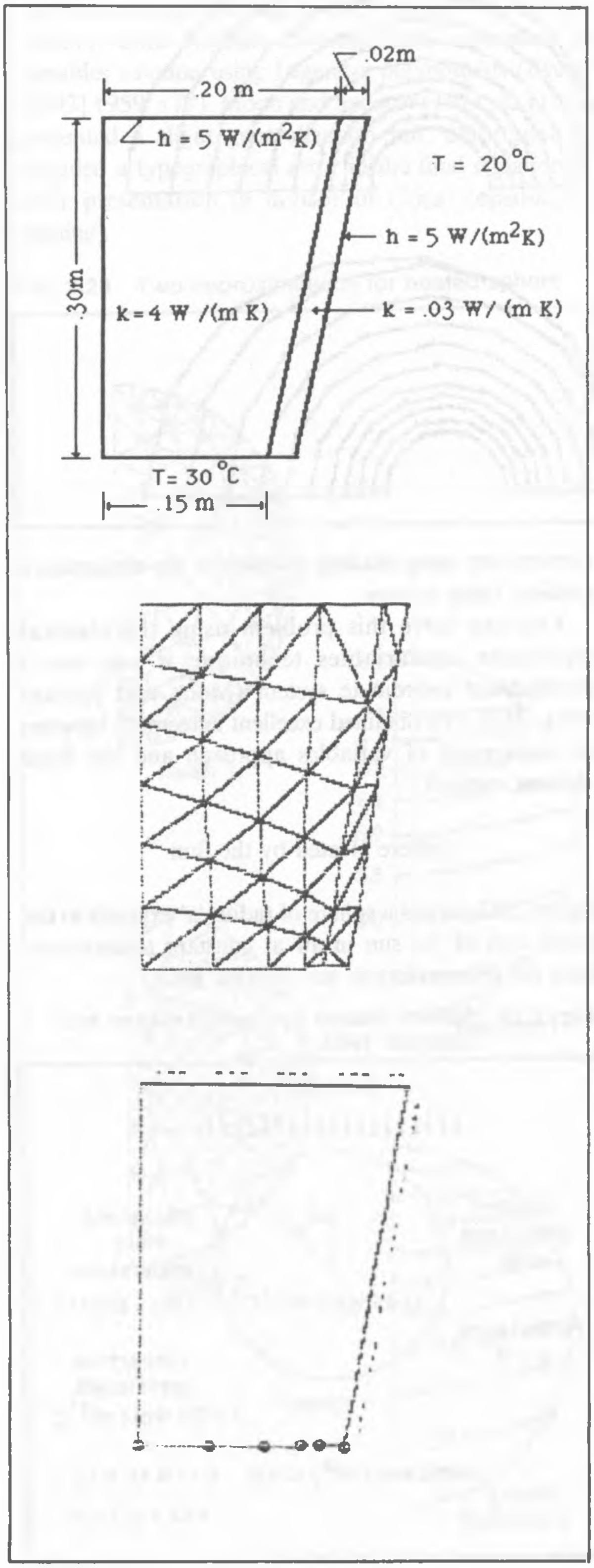
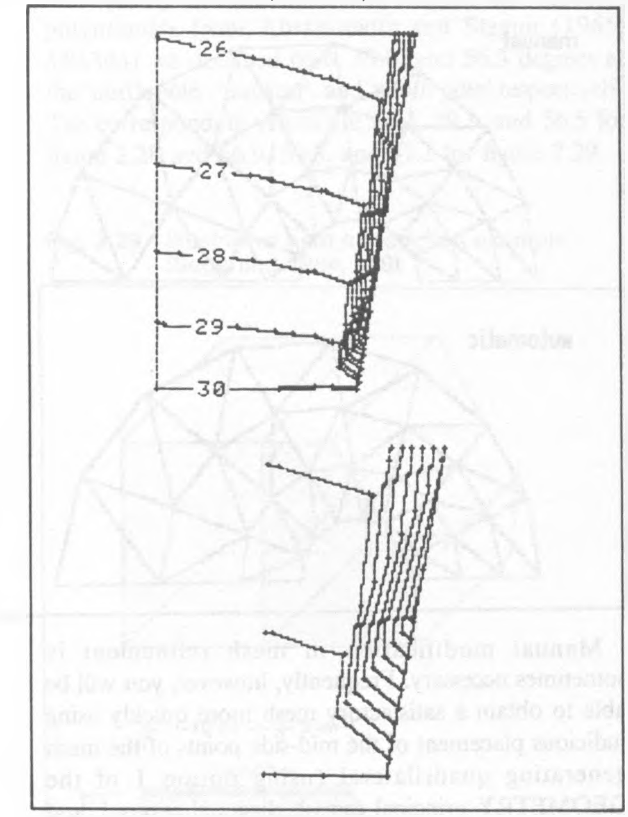


Fig. 2.23 Bottom-heated pot with insulated side (nodal temperature)



Eccentric Cylinder (Segerlind cover)

Figure 2.24 sketches the eccentric cylinder problem from Segerlind (1984, 405-410). The inner surface is heated uniformly, and the outer surface has convection into a constant temperature surrounding fluid. Segerlind uses a quadratic quadrilateral element, but you may use the same mesh generating regions for the triangular elements used here. Figure 2.25 displays two alternative meshes. The first matches the figure on the cover and illustrates the flexibility of modifying (using option 2 of the GRID principal menu) an automatically generated mesh; the second figure was generated automatically. The generated contour lines are identical.

Fig. 2.24 Eccentric cylinder (cover figure)

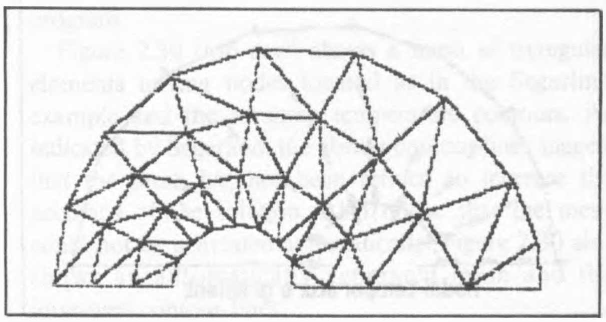
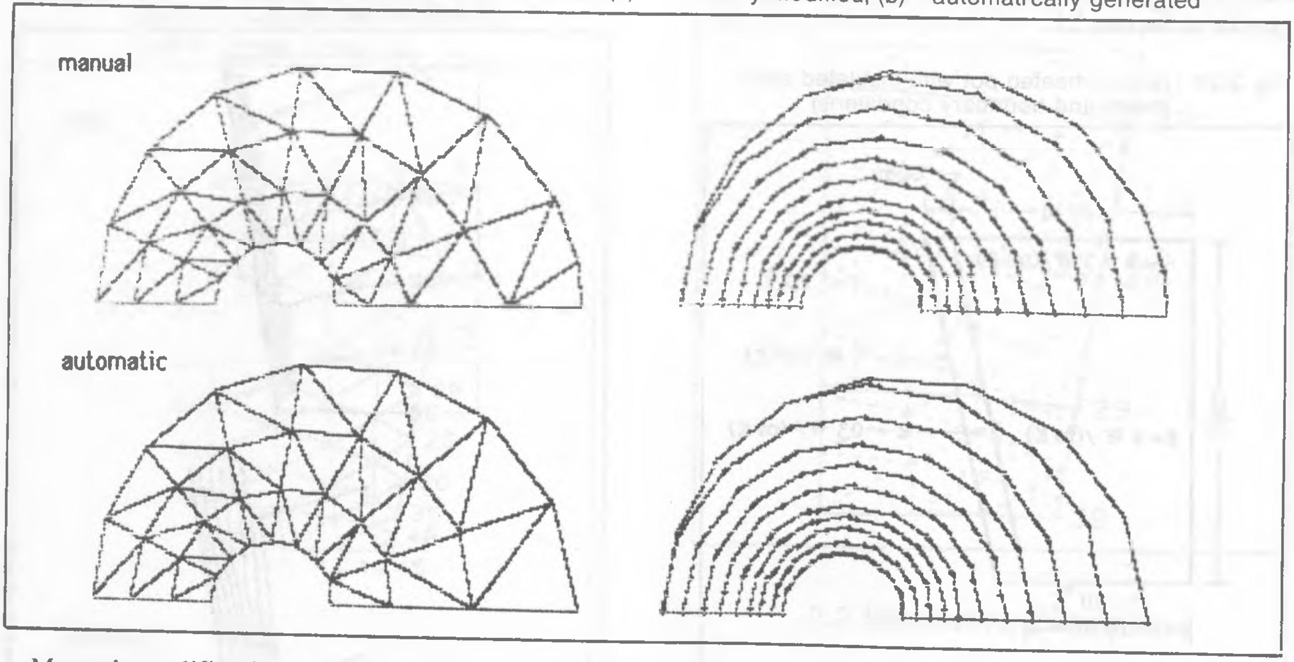
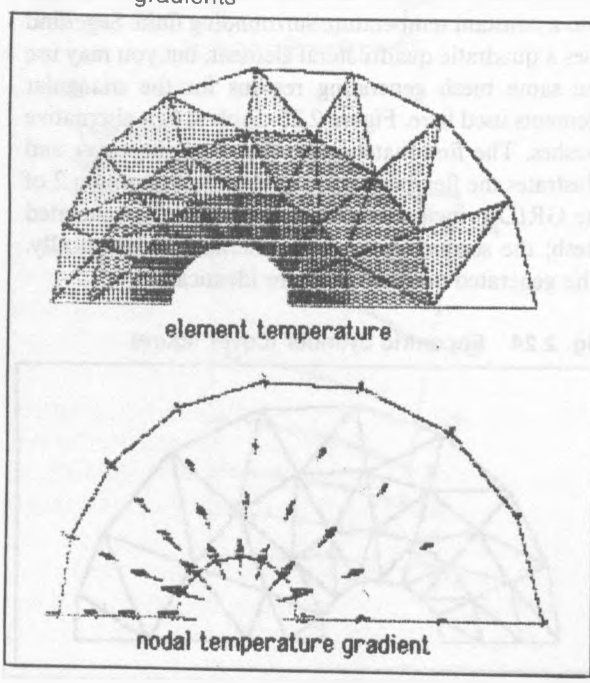


Fig. 2.25 Eccentric cylinder meshes and contours: (a) - manually modified; (b) - automatically generated



Manual modification in mesh refinement is sometimes necessary. Frequently, however, you will be able to obtain a satisfactory mesh more quickly using judicious placement of the mid-side points of the mesh generating quadrilateral (using option 1 of the GEOMETRY principal menu), diagonal reversal, and node dragging (using option 2 of the GRID principal menu). Figure 2.26 shows the average element temperature using shading and shows the temperature gradient using arrows.

Fig. 2.26 Eccentric cylinder average element temperature and nodal temperature gradients

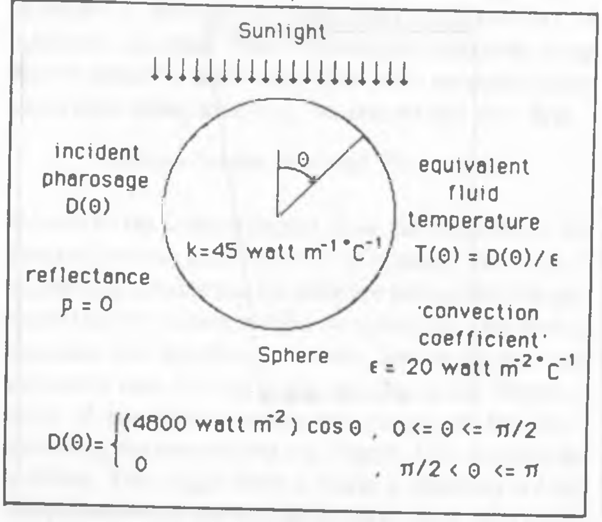


You can solve this problem using the classical separation of variables technique if you use a bicylindrical coordinate system (Moon and Spencer 1961, 365). We obtained excellent agreement between the separation of variables approach and the finite element method.

Sphere Heated by the Sun

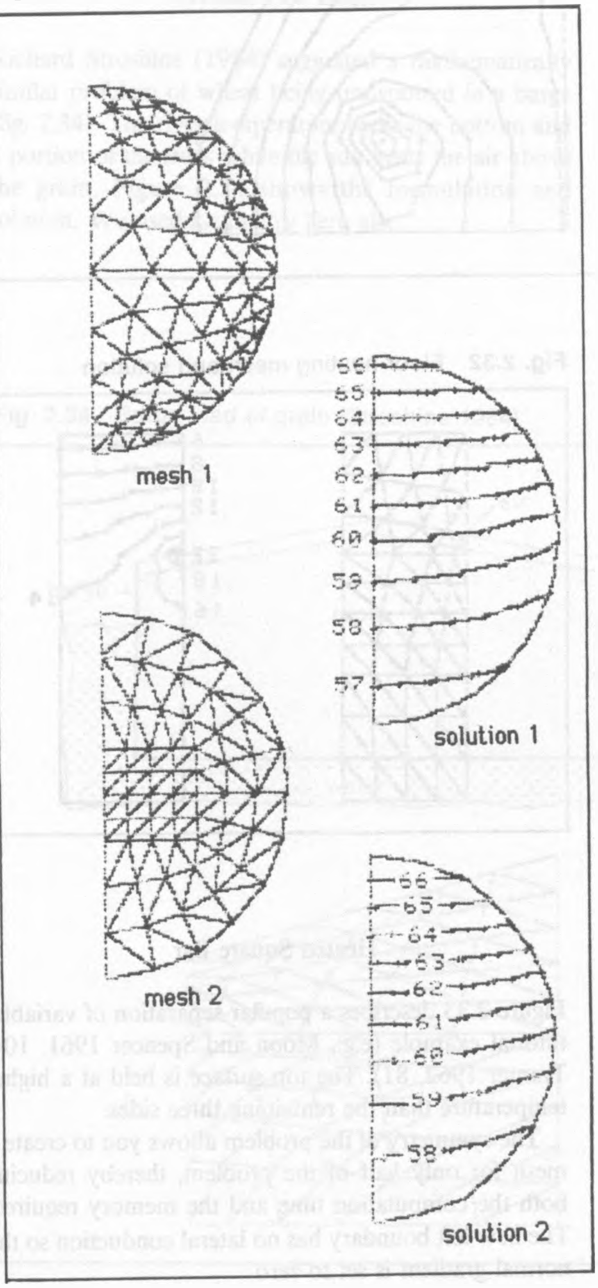
Figure 2.27 depicts a sphere of radius ‘a’ exposed to the direct rays of the sun in air at constant temperature. Find the temperature at any internal point.

Fig. 2.27 Sphere heated by sunlight (Moon and Spencer 1961)



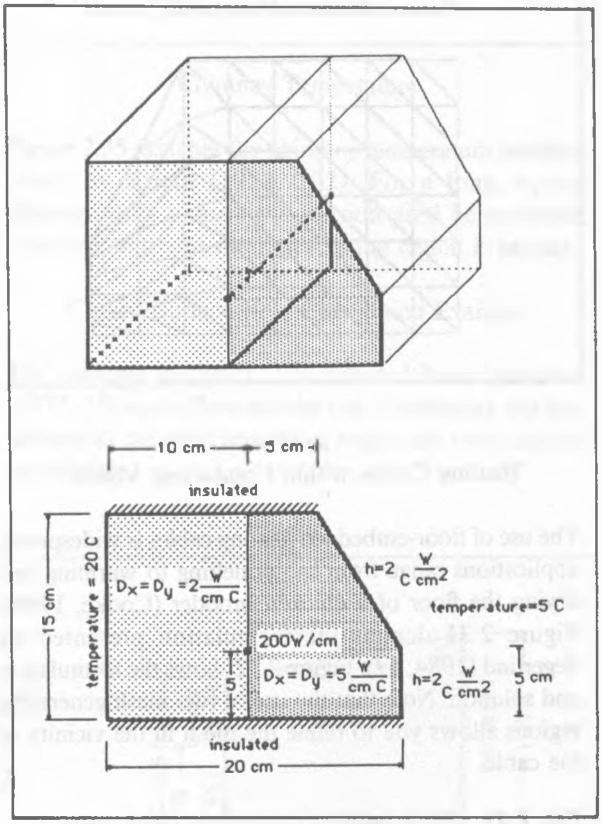
We formulated the problem two different ways using two alternative meshes (fig. 2.28). We formed the first with two degenerate sides of the quadrilateral in each region along the curved boundary. We varied the fluid temperature from the top to the 'equator'. In the last century, Lord Raleigh developed the separation of variables solution using Legendre polynomials (Byerly [1893] 1959, 177). Moon and Spencer (1971, 221) also presented a slight generalization but, unfortunately, included a typographical error in the final equation of their presentation [a divisor of (2)(a) (epsilon) is missing].

Fig. 2.28 Two approximations for heated sphere



Using the general expression for the series coefficient from Byerly and the recurrence relation for Legendre polynomials from Abramowitz and Stegun (1965, 340,344) we obtained 66.6, 59.0, and 56.3 degrees at the 'north pole', 'equator', and 'south pole' respectively. The corresponding values are 66.4, 59.1, and 56.5 for figure 2.28 and 66.9, 59.8, and 57.1 for figure 2.29.

Fig. 2.29 Illustrative heat conduction example (Segerlind 1984, 219)

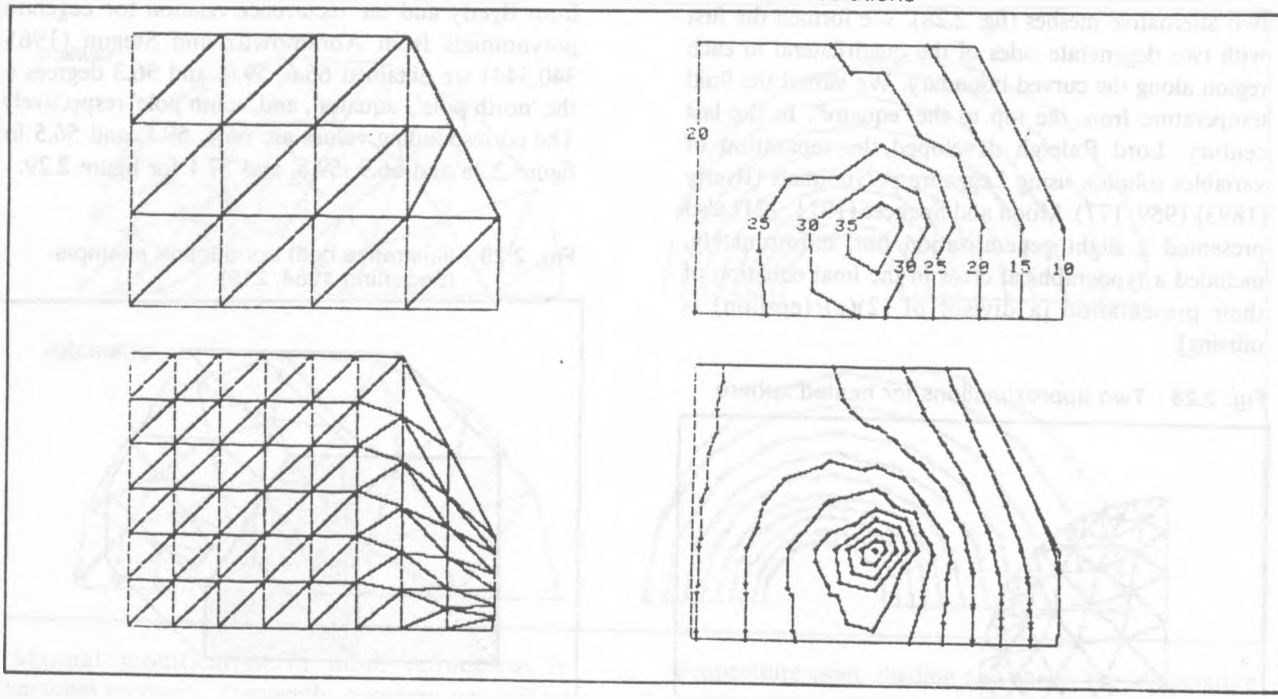


Segerlind's Illustrative Example

Segerlind (1984, 219-223) presented an example problem (fig. 2.29) to illustrate all of the possible data inputs for the two-dimensional heat conduction program.

Figure 2.30 (top row) shows a mesh of triangular elements having nodes located as in the Segerlind example and the resulting temperature contours. As indicated by Segerlind, the abrupt contour lines suggest that the mesh has not been refined to increase the accuracy of the solution. Also notice that the mesh could not be generated automatically. Figure 2.30 also shows an automatically generated mesh and the improved contour lines.

Fig. 2.30 Manually modified and automatically generated meshes and solutions



Heating Cables within Conducting Media

The use of floor-embedded heating cables is widespread; applications range from snow melting to warming and drying the floor of a chicken brooder (Cooke, 1969). Figure 2.31 depicts the formulation presented by Segerlind (1984, 49). Figure 2.32 shows the formulation and solution. Note that the use of two mesh generating regions allows you to refine the mesh in the vicinity of the cable.

Fig. 2.31 Floor heating (Segerlind 1984, 148)

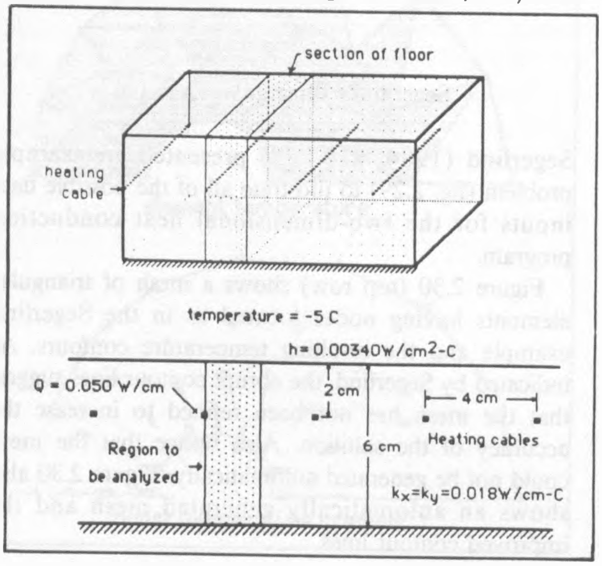
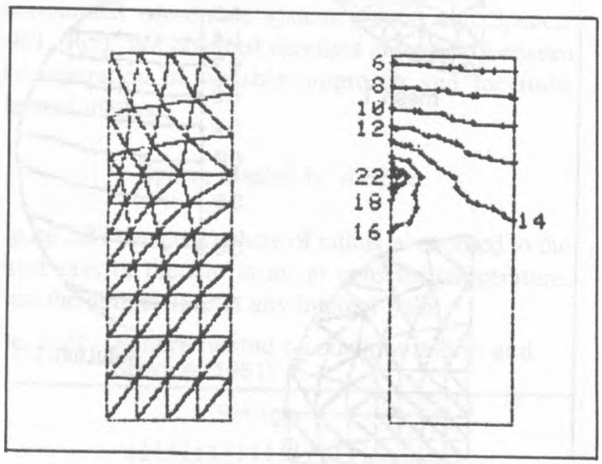


Fig. 2.32 Floor heating mesh and solution

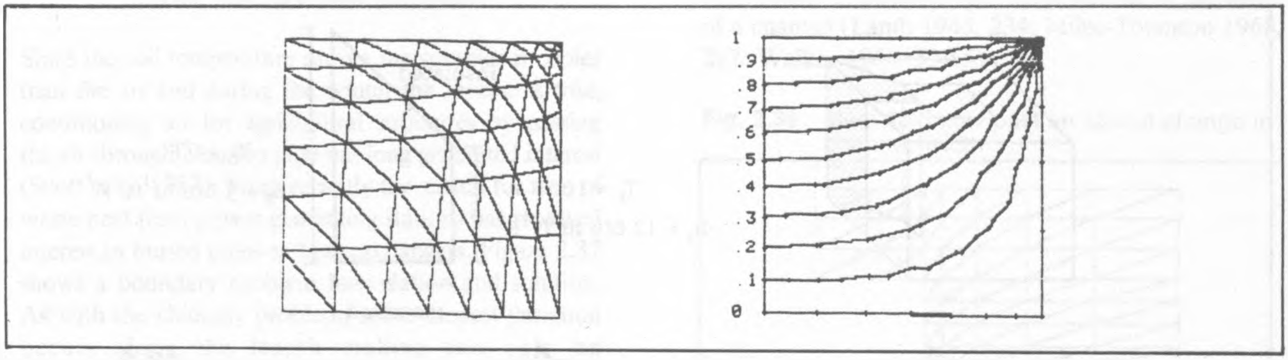


Heated Square Bar

Figure 2.33 describes a popular separation of variables tutorial example (e.g., Moon and Spencer 1961, 104; Tranter 1962, 81). The top surface is held at a higher temperature than the remaining three sides.

The symmetry of the problem allows you to create a mesh for only half of the problem, thereby reducing both the computation time and the memory required. The new left boundary has no lateral conduction so the normal gradient is set to zero.

Fig. 2.33 Heated square bar



Wheat In a Barge

Richard Stroshine (1984) suggested a mathematically similar problem of wheat being transported in a barge (fig. 2.34). The river temperature cools the bottom and a portion of the sides while the sun heats the air above the grain. Figure 2.34 shows the formulation and solution. We used symmetry here also.

Chimney Temperature

Figure 2.35 sketches the chimney temperature problem posed by Krieth (1958, 102). For a long, square chimney, only a one eighth-section need be modelled; therefore, only one mesh generating region is needed.

Curved Surface Heat Conduction Example

The example in Figure 2.36 adapted from Gustafson (1977, 33) again illustrates the use of symmetry but also shows that the mesh generating region can have circular boundaries.

Fig. 2.34 Barge load of grain (Stroshine 1984)

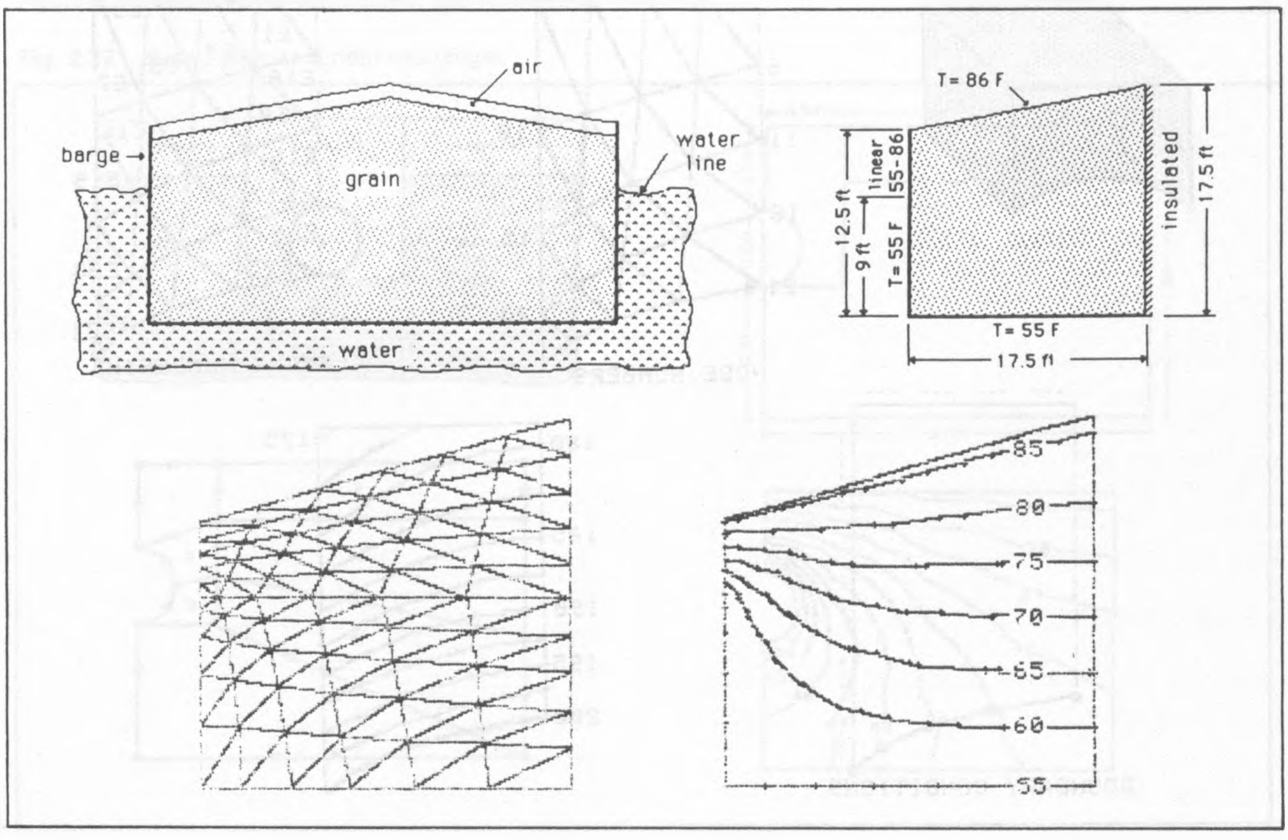


Fig. 2.35 Chimney (Kreith 1958, 102)

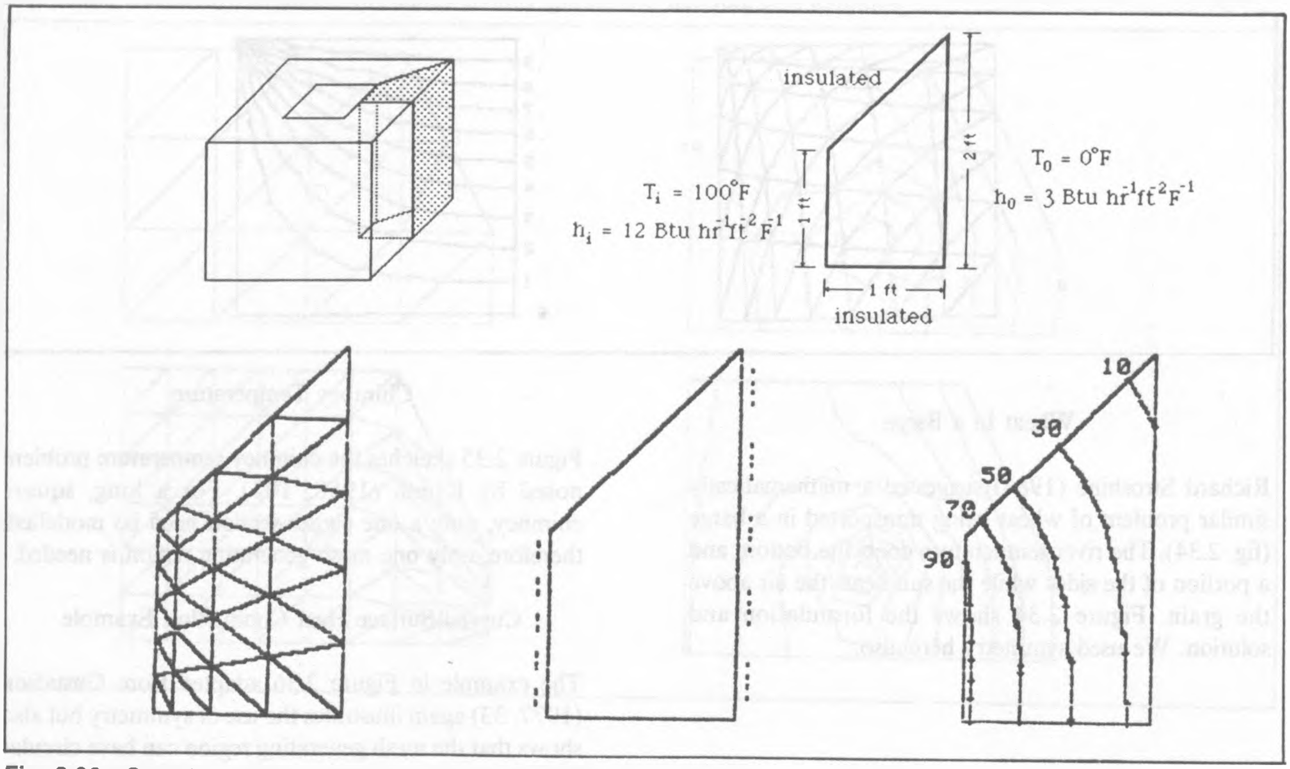
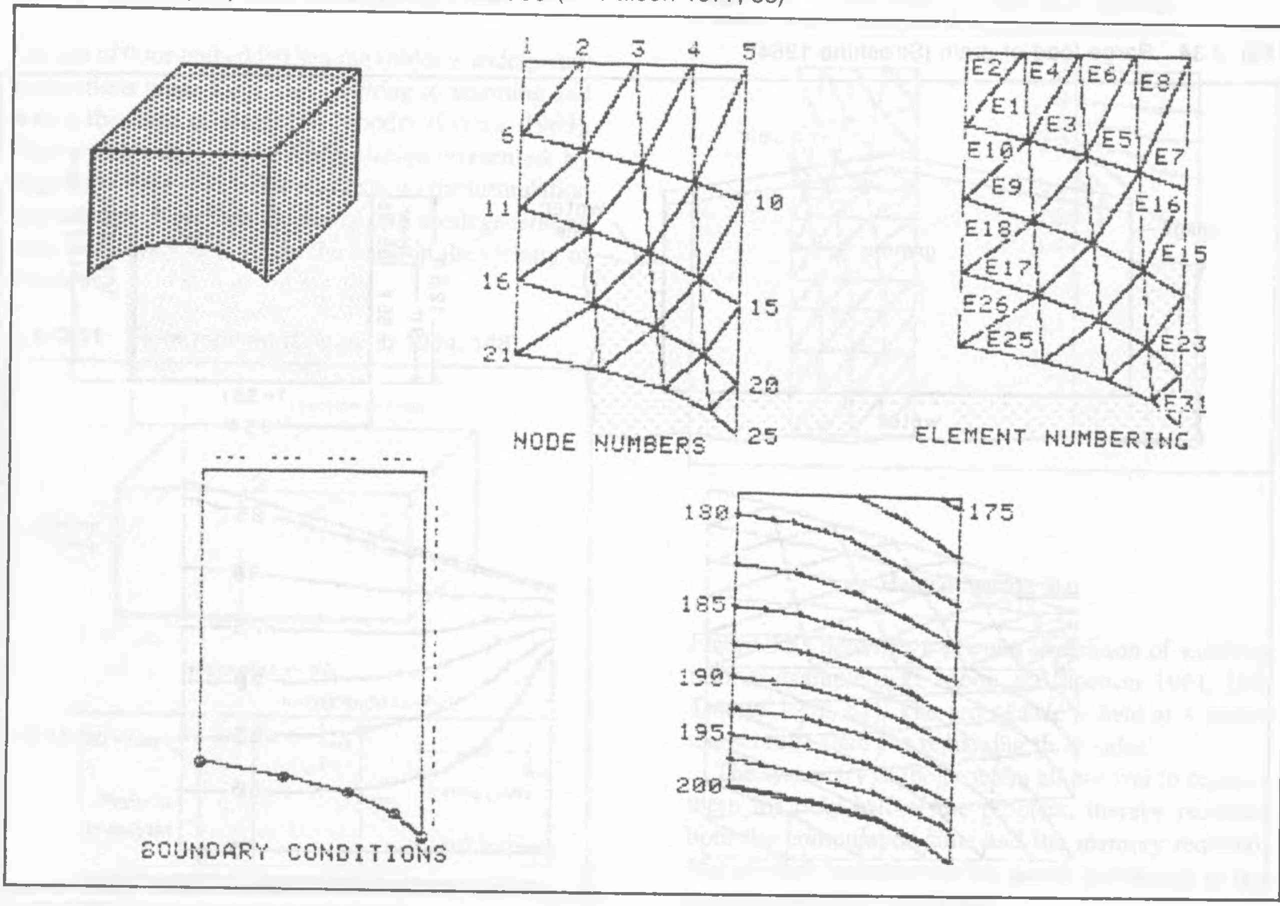


Fig. 2.36 Sample problem with curved surface (Gustafson 1977, 33)



Buried Pipe as a Heat Exchanger

Since the soil temperature during the summer is cooler than the air and during the winter the reverse is true, conditioning air for agricultural structures by passing the air through a buried pipe has long generated interest (Scott et al. 1983). More recently the search for uses of waste heat from power generating stations has renewed interest in buried pipes as heat exchangers. Figure 2.37 shows a boundary problem formulation and solution. As with the chimney problem, some modest variation occurs along the length making this only an approximate two-dimensional problem. This problem approaches the limit of the available memory.

2.2.1 Ideal Fluid Flow

Hydrodynamics abounds with examples of the use of Laplace's equation (Lamb 1945; Milne-Thomson 1968; Batchelor 1967). Velocity potential is the quantity analogous to temperature in heat conduction. The theory of complex variables (and conformal mapping) is also a rich source of problems (Rothe 1933; Kober 1957; Walker 1964; Moon and Spencer 1971).

The Abrupt Change in the Width of a Channel

Figure 2.38 describes the formulation and solution of the classical problem of the abrupt change in the width of a channel (Lamb 1945, 234; Milne-Thomson 1968, 287; Walker 1964, 53-65).

Fig. 2.38 Ideal fluid flow past an abrupt change in channel width

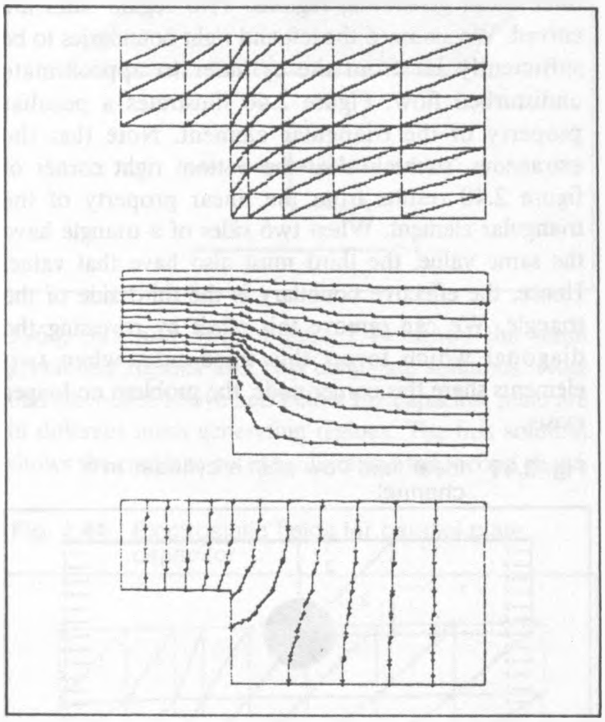
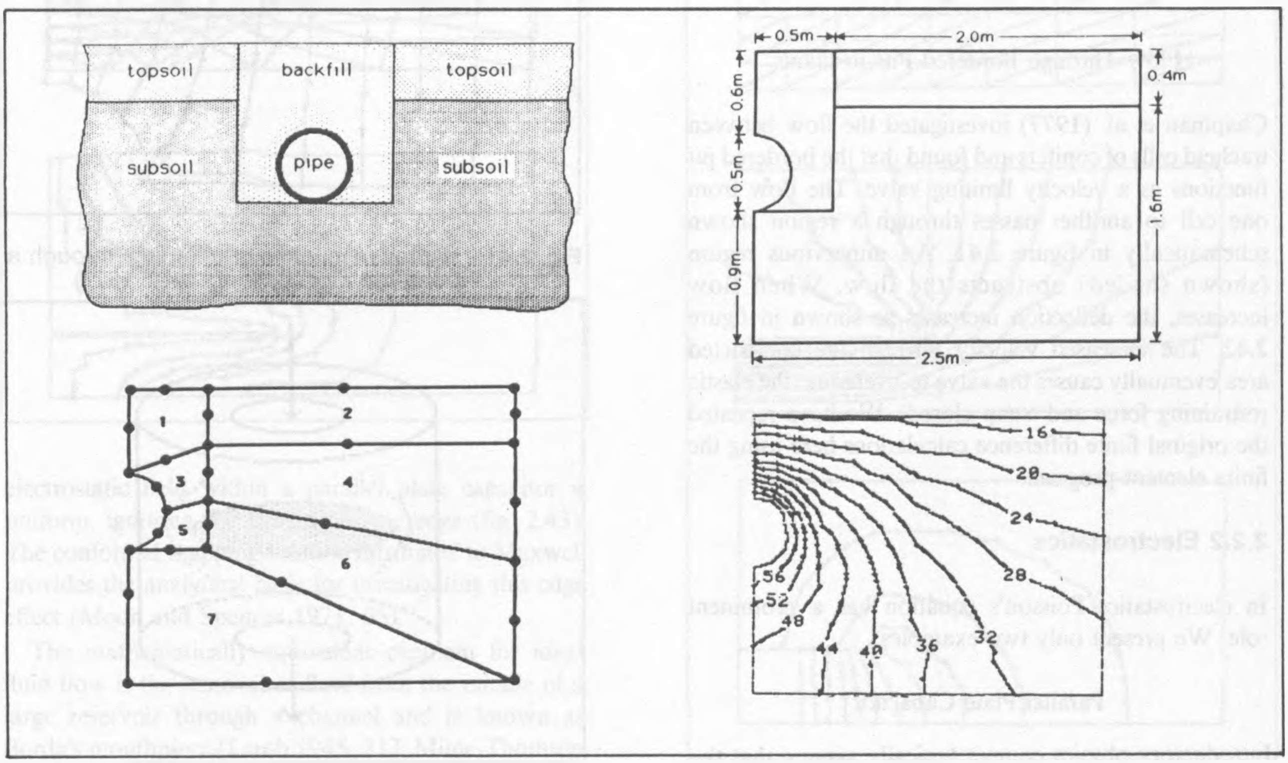


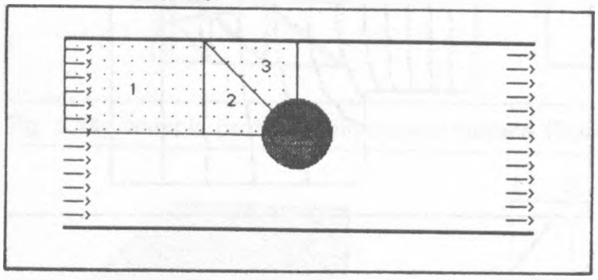
Fig. 2.37 Buried pipe as a heat exchanger



Ideal Fluid Flow Around a Cylinder in a Channel

Flow past a cylinder in a channel (Segerlind 1984, 126; Martin and Carey 1973; Lamb 1945, 212) can be handled easily by the program. In figure 2.39 we used three mesh generating regions. Two region sides are curved. We assumed the left and right boundaries to be sufficiently far from the cylinder to approximate undisturbed flow. Figure 2.40 illustrates a peculiar property of the triangular element. Note that the extraneous 'streamline' at the bottom right corner of figure 2.40 results from the linear property of the triangular element. When two sides of a triangle have the same value, the third must also have that value. Hence, the effective boundary is the third side of the triangle. We can remove this effect by reversing the diagonal which forms this third side; when two elements share this corner node, the problem no longer exists.

Fig. 2.39 Ideal fluid flow past a cylinder in a channel



Flow Through Bordered Pits in Plants

Chapman et al. (1977) investigated the flow between tracheid cells of conifers and found that the bordered pit functions as a velocity limiting valve. The flow from one cell to another passes through a region shown schematically in figure 2.41. An impervious region (shown shaded) obstructs the flow. When flow increases, the deflection increases as shown in figure 2.42. The increased velocity through the constricted area eventually causes the valve to overcome the elastic restraining force and 'snap closed'. We have repeated the original finite difference calculations here using the finite element program.

2.2.2 Electrostatics

In electrostatics Poisson's equation has a prominent role. We present only two examples.

Parallel Plate Capacitor

Introductory physics courses typically assume that the

Fig. 2.40 Streamlines and velocity potential lines for channel flow past a cylinder. (See text for discussion of the exceptional streamline.)

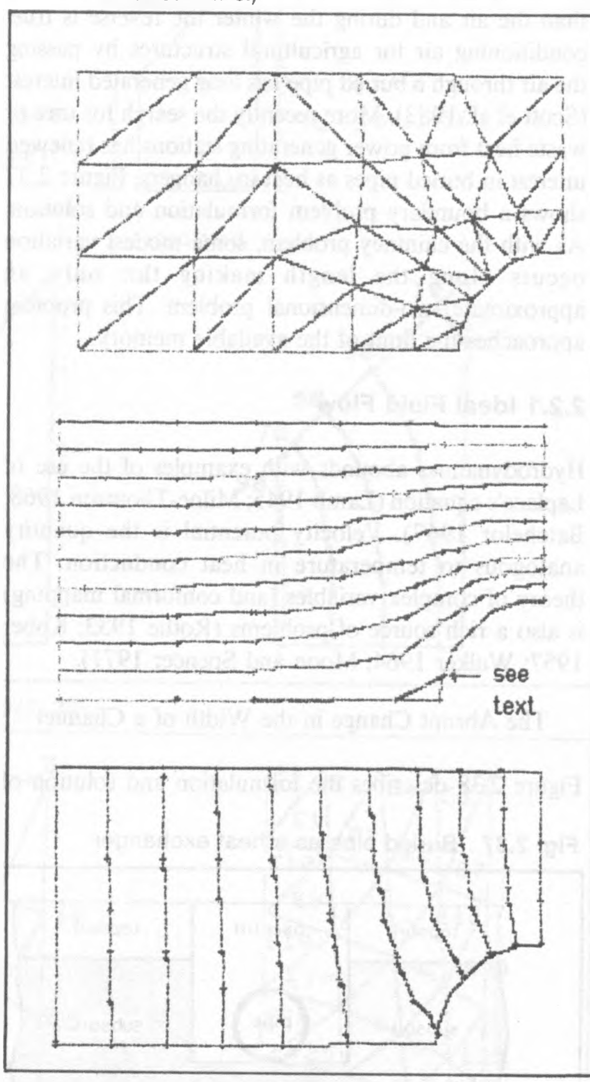


Fig. 2.41 Hydrodynamical model of flow through a bordered pit (Chapman et al. 1977)

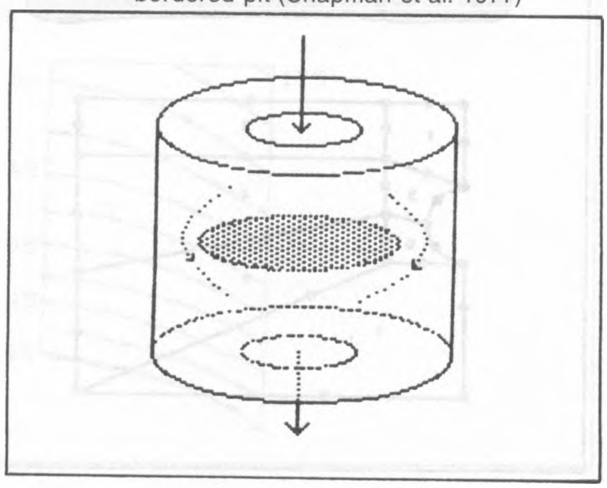


Fig. 2.42 Meshes and streamlines for two bordered pit configurations

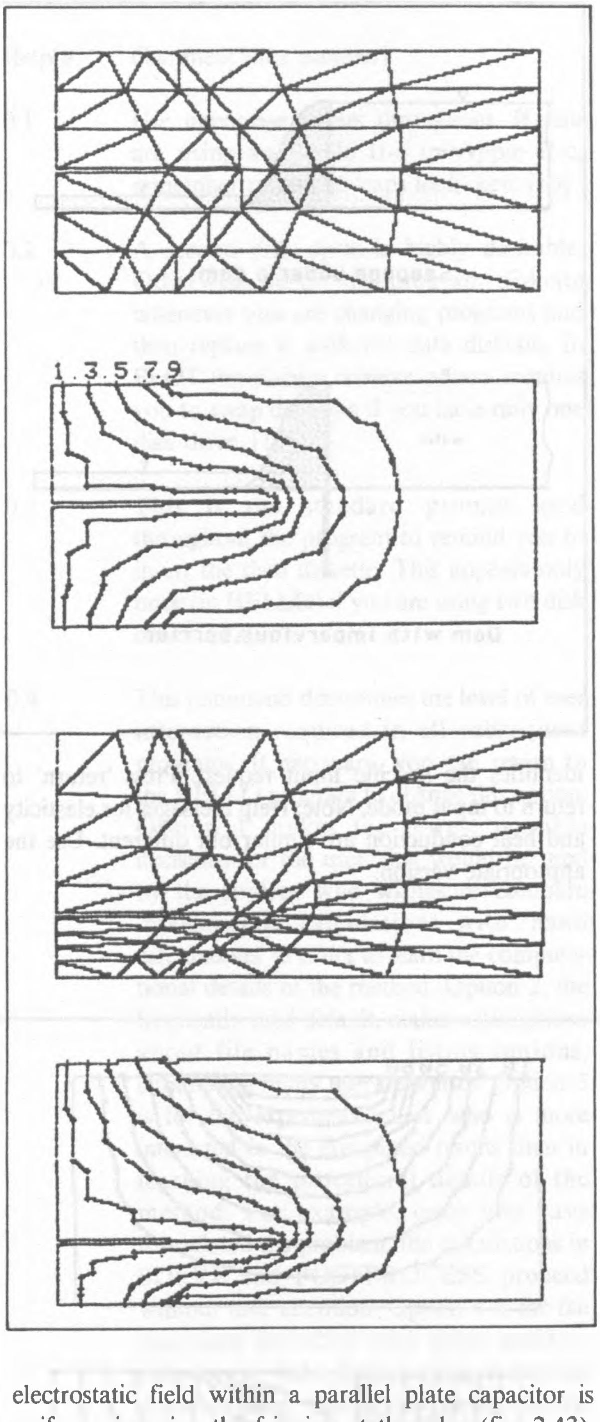
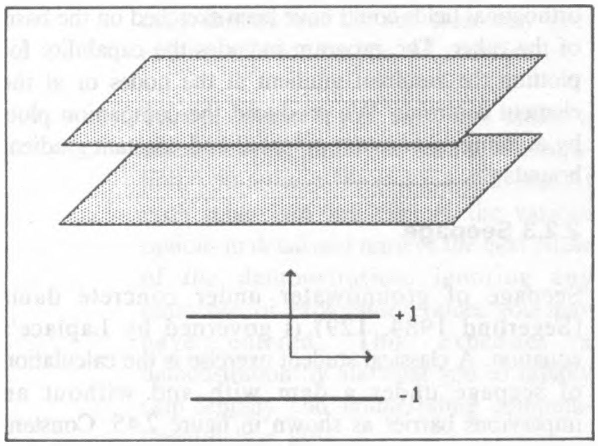


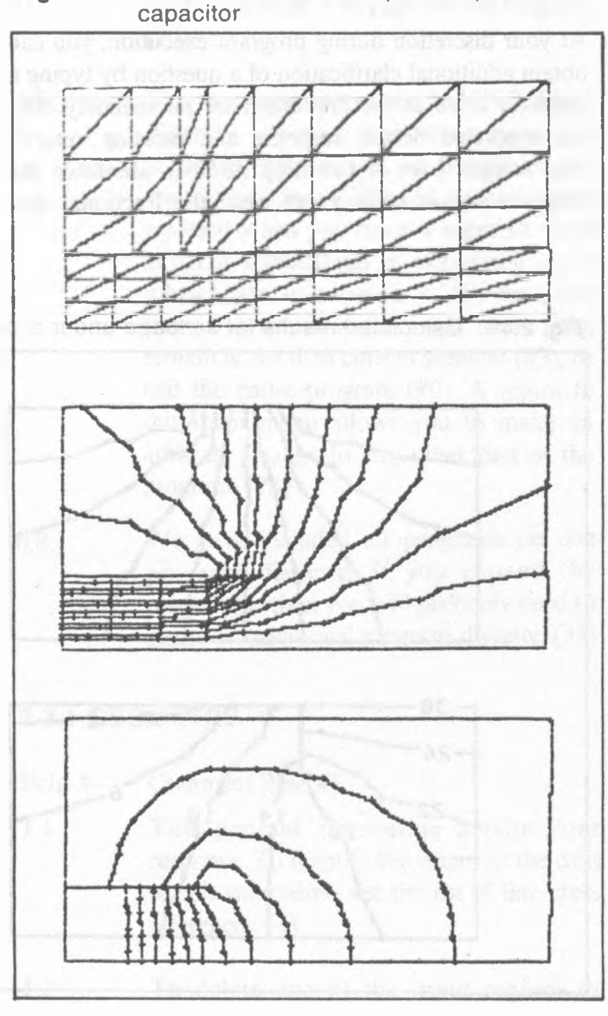
Fig. 2.43 Parallel plate capacitor



electrostatic field within a parallel plate capacitor is uniform, ignoring the fringing at the edge (fig. 2.43). The conformal mapping solution attributed to Maxwell provides the analytical basis for investigating this edge effect (Moon and Spencer 1971, 65).

The mathematically equivalent problem for ideal fluid flow is the removal of fluid from the middle of a large reservoir through a channel and is known as Borda's mouthpiece (Lamb 1945, 217; Milne-Thomson 1968, 307 and 331). Figure 2.44 shows the mesh generating regions and two conjugate solutions. Note that the nodes above and below the capacitor plate are in different mesh generating regions. The first solution shows the constant potential lines and the second shows

Fig. 2.44 Electrostatic fields for parallel plate capacitor



the electric field intensity lines. Either of these mutually orthogonal fields could have been sketched on the basis of the other. The program includes the capability for plotting the resultant gradient at the nodes or at the element centroids. We produced the companion plots by exchanging constant potential and constant gradient boundaries.

2.2.3 Seepage

Seepage of groundwater under concrete dams (Segerlind 1984, 129) is governed by Laplace's equation. A classical student exercise is the calculation of seepage under a dam with and without an impervious barrier as shown in figure 2.45. Constant piezometric potential lines for the two cases are shown in figure 2.46. As with the parallel plate capacitor problem, note that the mesh generating regions on the two sides of the barrier are not joined.

2.3 Help Messages for Heat Conduction

At your discretion during program execution, you can obtain additional clarification of a question by typing a question mark as the first character immediately after the bracketed default response and pressing 'return'. The integer part of the help number identifies the program segment being executed; the fractional part identifies the specific input request. Press 'return' to return to input mode. Note: Help messages for elasticity and heat conduction are similar but different. Use the appropriate version.

Fig. 2.45 Seepage under a concrete dam (Segerlind 1984, 137)

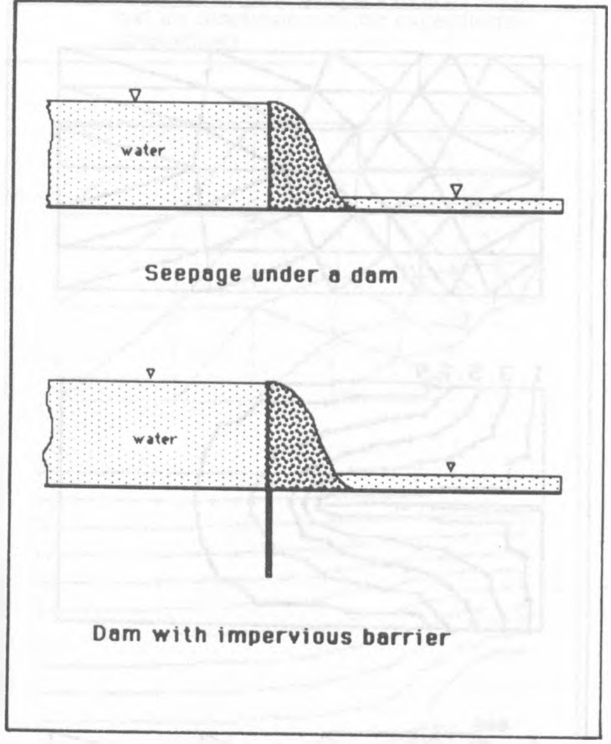
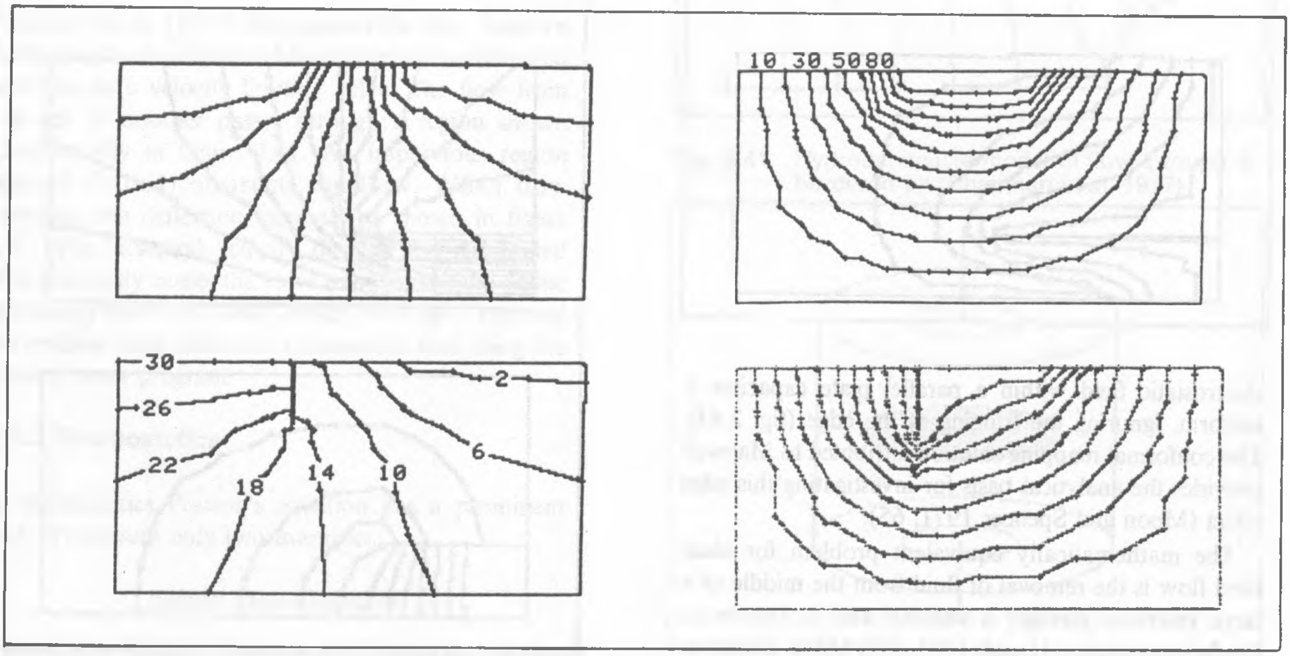


Fig. 2.46 Calculated results for seepage under a dam



2.3.0 Hello

Help #	Comment [line number]
0.1	Use uppercase letters throughout. If you are using an Apple II-e or Apple II-c, remember to set the 'caps lock' key. [10]
0.2	A second disk drive is highly desirable. Otherwise, insert the program diskette whenever you are changing programs and then replace it with the data diskette. In PLOT the picture retrieve option requires you to swap diskettes if you have only one disk drive. [12]
0.3	This is the standard prompt used throughout the program to remind you to insert the data diskette. This appears only once (in HELLO) if you are using two disk drives. [12]
0.4	This command determines the level of user interaction required in all subsequent programs. If necessary, you can return to the HELLO program to change this option. Option 1 makes the fewest possible decisions for the user; this would be used by the student who wishes to compare intermediate calculations with 'hand calculations' in order to learn the computational details of the method. Option 2, the frequently used default, makes assumptions about file names and listing options, eliminating many user responses. Option 3 is for the experienced user who is more interested in the computed results than in learning the procedural details of the method. For example, once you have formulated the problem, the calculations in SOLVE and POSTPROCESS proceed without user attention. Option 4 is for the classroom instructor who might accidentally erase data files during a lecture demonstration. This option suppresses the saving of any data, permitting the data diskette to be used with a 'write protect' tab.
	Normally, the program disables any pre-existing data files which might become inconsistent with the immediately entered data. For example, if you alter the coordinates of a node, then you automatically disable but do not erase any pre-existing solution for the old coordinates.
	To use option 4 you must first create a demonstration problem. In a demonstration, you retrieve the existing data file. At each stage you can explore the various options in detail and retrieve the next phase of the demonstration, ignoring any erroneous or exploratory values you may have entered. This expedites a demonstration by allowing you to rapidly skip lengthy and uninteresting computational delays. [14]
0.5	Review option 4 above (0.4).
	As a safety measure you must deliberately over-ride the default to enter this mode. [16]
0.6	This is the main title page for the program. [20]
0.7	This is referred to as the MAIN MENU. Normal program execution proceeds sequentially down the list of options, and the default menu item is automatically updated when you finish a segment. Each program segment has an exit option which allows you to proceed to the next step (item #1), return to this main menu (#2), remain in the then current segment (#3), or exit the entire program (#0). A return to this main menu allows you to make an arbitrary change to any other part of the program. [28]
0.8	We have included all programs on one program diskette. If you expand the program further, you will probably need to create an additional program diskette. [38]

2.3.1 GEOMETRY

Help #	Comment [line #]
1.1	This general subroutine awaits your response. To identify the origin of the calls to this subroutine, see the list of line cross references. [4]
1.2	To delete one of the input regions (a

curvilinear quadrilateral), specify the region by number. [40]

1.3 Each curvilinear quadrilateral used to generate a mesh of triangular elements can have from 2 to 9 element nodes (including the two vertices on each side of the quadrilateral but not the mid-side point). After specifying the number for this side, either select another side (use arrows and 'S') or, if you have specified all sides, select 'Q' to quit. Do not exit until you have specified all sides. [54]

1.4,1.5 This defines the ending point for a line or arc segment; the point must be within the range of the global coordinate system. [60]

1.6,1.7 Specify the (X,Y) or (R,Z) coordinates for the ending point of the arc or straight line which you use to generate points that will define quadrilateral input region(s). [72,74]

1.8,1.9 A circular arc generates points needed to define a quadrilateral input region. Specify the center of the arc. [80,82]

1.10,1.11 Specify the (X,Y) or (R,Z) coordinates for the beginning point of an arc or straight line which you use to define quadrilateral input regions. [86,88]

1.12 An arc generates points to define a quadrilateral input region. The signed angle is in degrees; counterclockwise is positive. [80,82]

1.13 The number of INTERMEDIATE points on an arc or a line does NOT include the beginning and ending points (unlike the question about the number of nodes per side). [96]

1.14,1.15 Evenly spaced points produce equal-sized elements. The mesh should be finer in areas of greatest change and coarser in areas of smallest change. Therefore, adjust the placement of the mid-side nodes to achieve this effect. Usually a value of 0.3 to 0.7 is workable. If you place the mid-side nodes too near the extremities, the algorithm will fail (see Segerlind, 369 or Steinmueller 1974). [96,100]

1.16 By pressing 'E' or 'D' enter or delete the coordinates of points to be used to define the mesh-generating input regions. The program checks for duplicate points. Use 'Q' or '>' to quit or use '<' to branch to the menu (1.05). [108]

1.17
1.16 When you have selected 'E' above, you can enter a point, a straight line, or an arc; quit; or branch back one question. [110]

1.18 To begin creating a region, you must specify its number. Then you 'point' to the eight defining points in a counterclockwise order. [118]

1.19 You cannot exit until you have specified all eight points needed to define the region. After defining all eight points, select 'Q' to quit and visually verify that you have formed the region in a counterclockwise sequence. This is a vital user check. [124]

1.20 This gives you an opportunity to leave the region generation process when you have defined all regions. [128]

1.21 You can add or delete regions. Press 'E' to enter a region, 'D' to delete a region, or 'Q' to quit. [130]

1.22,1.23 Without graphics, you can enter points (X,Y) used to define the mesh generating regions. '>' branches out of the input mode and '<' returns you to a previous point. [152]

1.24,1.25 To modify any of the coordinates, you are asked for the starting point number. [154]

1.26,1.27 Select 8 points from the previously entered list of coordinate pairs (without graphics) to form a region in counterclockwise order. [158,160]

1.28,1.29 Here, as throughout the program, you can revise your previous input. [162]

1.30,1.31 Specify the number of nodes for each pair of opposite sides. Without graphics, you must remember to match the values for regions having a common boundary. [166]

1.32,1.33 As throughout, you have an opportunity to modify previously entered data. [168]

1.34 The edit menu allows you to revise any of the preceding steps (input coordinates, selection of region points, and designation of the number of element nodes to be placed on each side of the mesh generating region). [172]

1.35 Here you can edit the mesh generating regions. [178]

1.36 When you save a data file, specify the file name to be used. Unless you need several alternative problem formulations, accept the default. The free-run mode uses the default names automatically without additional user action. [226]

1.37 When retrieving a data file, specify the file name. The free-run mode uses the default name. [234]

1.38,1.39 Error traps allow you to correct disk problems. [258,260]

1.40 This is the only opportunity for you to choose between a two-dimensional and an axisymmetrical geometry. The element cross-section still appears to be triangular in either case; however, in the axisymmetrical case the triangle sweeps out an annular region. Mathematically the problems are similar but different (Segerlind 1984, 87-99 and 165-176). Likewise, the plots produced are similar in appearance but different. For axisymmetric problems, you must use the Z axis as the axis of symmetry (see Segerlind fig. 13.1). [WARNING: This question is only asked during a cold start so do NOT use the control Q warm restart command here, bypassing the answer to this question.] [274]

1.41 This program establishes coordinate points that are used to define quadrilateral regions used in the following GRID program to generate the discretized mesh. You need to use judgment in creating this mesh because this step influences the accuracy of the solution.

The principal menu for each program is numbered (X.05), as here. Because data input with graphical support is both easier to use and less prone to undetected errors, you will usually elect options 1 and 3 in that order. Nevertheless, data input with a closer correspondence to the batch programs of Segerlind is possible with options 2 and 4.

Here, as in subsequent programs, you can review the generated data that is stored immediately on disk. Both a tabular listing and a graphical representation are possible. During listings, the output is presented by the screenful. Press 'return' for the next screen; press 'space' for the next line; or press a number, 0-9, to scroll the data. [284]

1.42 Each program segment has an exit menu. The first option leads you to the next logical step in the solution and is the default that can be accepted by simply pressing 'return'. Should you need to branch to a different program segment or to change the user interaction default, the menu allows you to return to the HELLO program (see Help 0.4).

Throughout the program we have attempted to give you an opportunity for second thoughts. Here, that means you can return to the current program. You can also use the 'less than' symbol, '<', with 'return' to go back to a previous step if you type it as the first character after the bracketed default prompt.

To keep you from entering the middle of a FOR-NEXT loop, etc., the 'branch backward' option sometimes sends you back further than you might have expected. Sometimes you will find it even harder to anticipate the action of the 'branch forward' command '>', used when you wish to skip a section. (Note: The variable BC% is used consistently throughout the source code to route the branching.)

We have placed the termination option last throughout and have numbered it '0'. [286]

1.43 If you are reviewing a problem in demo

mode (Help 0.4) or are revising a previous solution, retrieve the coordinate data (see Help 1.41 - the data being retrieved was created with option 1 or 2 of menu 1.05). If retrieved, the defaults presented reflect the retrieved data; a sequence of 'return's allows you to quickly advance to the position you wish to alter.
To change the number of nodes on a side of the mesh generating region, a frequent modification, go immediately to option 3 or 4, skipping options 1 and 2 of menu 1.05. [294]

1.44 Here you can retrieve the region data from a previous exercise (see Help 1.43). This data includes the connectivity of the 8 points defining each region and the number of element nodes to be placed on each side (including the vertices). [296]

1.45-1.48 To make a global plot of the problem including all regions, specify the limits of the coordinate axes. You exclude coordinates of points outside this region. [298,300]

2.3.2 GRID

Help #	Comment [line number]
2.1	This general purpose subroutine allows you to decide when to proceed. Refer to the list of line cross-references to determine the origin of calls to this subroutine. [4]
2.2,2.3	This code allows you to relocate a node. Enter the new (X,Y) or (R,Z) coordinates. [18]
2.4,2.5	Use the graphics cursor to point to individual nodes and elements. Begin the sequential review at any node or element. Use the left and right arrows to move through the list and press 'Q' to quit. After leaving node identification, begin element identification using the same commands. [190,194]
2.6	If you wish to retain only one boundary value problem on the diskette (the usual case), use the default file names by pressing 'return'. If you wish to retain several variations in the problem, to prevent the over-writing of data you must use a different file name in saving the data. [Note: you must remember any alternate names for subsequent retrieval (see Help 2.7)]. [204]
2.7	Here as in Help 2.6, specify the name of a data file for retrieval. The free-run mode (see user interaction level Help 0.4) automatically selects the defaults for Help 2.6 and 2.7. [214]
2.8,2.9	These are two error recovery options. You can unlock a data file (necessary if you wish to replace the data using the same file name), or you can review the existing list of file names using the catalog command. [234,236]
2.10	This is the principal grid definition menu (2.05). Characteristically, you proceed sequentially through the list. Option 1 allows you to handle the process with fewer user responses provided you wish to execute options 2-4. If you are exploring the computational details, you may wish to proceed more deliberately. Consistently throughout the programs, you have an opportunity to immediately review the results (options 5-7). Select option 0 if you are ready to leave this program (i.e., go to the exit menu). [258]
2.11	The exit menu has the standard options - proceed to the next logical program segment, return to the menu in the HELLO program for arbitrary branching, return to the current program, or leave the program entirely. The single-disk-drive user must swap diskettes. [262]
2.12	This option appears only if you are not in free-run mode and allows you to decide the level of interaction required in this program. If you are studying the methodology, you may want access to more computational details. [270]
2.13	This is menu option 2 and generates the mesh nodes and elements. You can (1) generate the grid automatically using the data entered in the GEOMETRY program,

(2) enter the data without graphics support (necessary if you are creating specialized meshes such as Segerlind's figure 11.8, figure 16.4), or (3) generate the cover page which cannot be generated automatically. Usually you can generate satisfactory meshes automatically. The third option is also useful if you want to generate the mesh automatically and then make minor adjustments. Make the modifications to elements and nodes at the end of the list.

Note that node relocation and diagonal reversal is handled easily elsewhere. [276]

2.14 The display option allows you to utilize plotting. Under the free-run option (global or in Help 2.12) this question defaults automatically to 'without user interaction'. [228]

2.15 The grid listing option also allows you to examine the details of the calculation process. Option 1 is usually preferred, deferring the listing to options 5-7 of menu 2.05 (see Help 2.10). [280]

2.16 In the free-run mode the grid plotting option default involves plotting without labels. If you label every node or element, the labels can become cluttered. After you have completed the solution, you have an opportunity to label selectively in PLOT.

The extensive use of graphics greatly diminishes your need to be aware of the element and node numbers. Usually you need to know only a few strategic node numbers. You can easily move a significant portion of the procedural details into the background. [282]

2.17 This is another option that you can use only when you are not in the free-run mode. Select option 2 only if you wish to explore the behavior of the renumbering algorithm.

By judiciously renumbering nodes, you can greatly expand the scope of problems that you can solve using this microcomputer. While execution time for this part of the program can become significant, the real benefit is a more compact formulation of the global equations, needed if significant instructional problems are to fit within the limited memory. The additional computation at this step also reduces execution time. The bandwidth reported here represents the maximum difference between the node numbers connected through a side of a triangular element plus one. This corresponds to the width of the array required to store the stiffness matrix if each node has one degree of freedom, as is the case here. Since the system of simultaneous linear algebraic equations is resident in memory during the Gauss elimination solution, the node renumbering process becomes a significant step. (Note 1: other solution techniques permit the solution if only parts of the large array are resident at any time. Note 2: The Collins renumbering algorithm does not guarantee the optimal renumbering. The original renumbering is retained if the bandwidth is not reduced.) [290]

2.18 In the initially generated mesh, each of the three sides of an element is plotted even when a side coincides with a previously plotted element. To avoid the redundant double plotting of common element boundaries and to identify the boundaries required in the PREPROCESS program, unique lines are identified. [296]

2.19 You can elect to list either the original or the revised node numbering. This is the first question in menu option 5. [308]

2.20,2.21 When you elect menu option 7 to plot the mesh and identify elements and nodes, you can select the original or renumbered nodes. [318]

2.22-2.25 When you wish to modify the mesh by moving nodes, enter the new coordinates here. [320]

2.24,2.25 To check the counterclockwise ordering of nodes in an element, specify a region. [322]

2.26 When you enter nodes and elements without graphics, you can retrieve the existing data file which can be treated as the default. This file may actually have been created using graphics. [336]

2.27,2.28 Because the program makes more efficient use of memory through dynamic dimensioning of arrays (i.e., arrays may be created and erased during execution), you need to specify the array size for just the memory required. If you are modifying existing data, note that the changes occur at the end of the list. Specifically, if you are reducing the array size, the existing array is truncated and copied into the new array. If you enlarge the array, append the new additions to the end. If you use the mesh generator to create the bulk of the mesh to be modified here, define the other regions such that you make changes in them at the end of the list. [338]

2.29 You can enter the coordinates of nodes, the global numbering of the nodes, or neither. Observe the counterclockwise enumeration of the nodes when you enter the global node numbers. [342]

2.30,2.31 The editing process permits you to enter values. Use '<' to decrement the counter and '>' to branch out 2.32 (i.e., end the editing). Start at a user specified index and specify the maximum subscript for the nodal coordinate array or array of node numbers for the save. [346,348]

2.33 You can elect to either adjust node positions or reverse diagonals. 'Well-shaped' elements give better computational results; you should avoid triangular slivers. If you move a node, the program erases and redraws the old elements. The mesh generating routine automatically selects the diagonal of an element pair that gives the better aspect ratio. Sometimes, however, you may wish to over-ride this (e.g., to obtain greater symmetry in the mesh or to avoid having two sides of an element fall in the corner of a boundary). [352]

2.3.3 Preprocess

Help # Comment [line number]

3.1 This general-use subroutine allows you to receive information and awaits your response before clearing the screen and continuing. See the line cross-reference listing to locate the calls to this subroutine. [4]

3.2 This optionally displays the list of files saved on the data diskette. [40]

3.3 During the entry of the material properties, you can specify an isotropic medium (i.e., you can set the conductivity in the direction of the X-axis equal to the conductivity in the Y-axis direction). Note that these directions coincide with the axis directions and in some instances may dictate orientation of the coordinate system. [66]

3.4 You can base data entry on three different categories - the entire body, the mesh generating region, or individual elements. When the property is a single constant throughout the entire body, enter that single value by selecting option 1. If you can characterize input (mesh generating) regions by a constant for each region, select option 2. Finally, if the properties vary from element to element, select option 3. Note: if the property is relatively uniform for the entire body but differs in a few places, choose option 1 and use option 2 or 3 to edit the initialized exceptions. [70]

3.5 Enter the uniform property value here. [72]

3.6,3.7 If a property can assume only a few values (up to 6), assign those values to letters and circumvent the repeated entry of multidigit numbers. When assigning the properties to the problem (see Help 3.8), either enter the letter for a coded value or over-ride the code by entering a numerical value. First specify the number of coded values being used and then supply the actual coding. [76,78]

3.8 Input the properties for each element. Enter a code, the actual number, or 'Q' to quit. [82]

3.9-3.11 As is customary, you have an opportunity to revise your input. Edit by region or by element. Specify the 3.11 beginning region or element number. [88,90]

3.12 Select the property to be edited. For axisymmetric problems, the main menu does not include the thickness property. [94]

3.13 Enter properties by region in the same manner that you enter elements (see Help 3.7). [102]

3.14 Enter or delete boundary conditions for a single node or for a range of nodes with a single type of boundary condition. Specify (1) the boundary condition type (e.g., nodal temperature on the boundary, surface flux, convection, or nodal heat source on the boundary or at an interior node), (2) the value(s) associated with a boundary condition (e.g., convection coefficient and fluid temperature), and (3) either a single node or a range of consecutive nodes around the boundary. When you specify a range of nodes, use a COUNTERCLOCKWISE traversal of the boundary to determine which is the beginning and the ending point. Use ‘E’ to enter a boundary condition and ‘D’ to delete. Use ‘Q’ to quit. The branching codes ‘<’ and ‘>’ take you back to the instructions or forward just as ‘Q’ does. [114]

3.15 Now select the boundary condition type to be entered or deleted. The four possibilities are (1) nodal temperature, (2) surface flux, (3) convection, and (4) nodal sources. ‘<’, ‘>’, and ‘Q’ all branch to the previous question (see Help 3.14). [116]

3.16 Convection may occur across the boundary that you see edgewise on the screen and is referred to as ‘Boundary (out)’. For a two-dimensional body, convection may occur through the front or back surfaces seen in plan view and is referred to as ‘Face (out)’. [120]

3.17 A surface flux boundary condition has four possible modes of application: (1) X-direction flux component; (2) Y-direction flux component; (3) Boundary (out), used to specify the normal component when the boundary (seen edgewise) does not coincide with one of the coordinate axis directions and is positive in the outward direction; and (4) Face (out) which corresponds to the front and back faces where the positive direction is outward. Note: If you desire convection at both front and back surfaces, use a composite surface conductance (convection coefficient) (e.g., double if the ends are the same.)

‘<’ branches to the logically previous question (Help 3.15) to allow you to select a different boundary condition type; ‘>’ branches one step further back (Help 3.14) to allow you to decide to enter, delete, or leave this option. [122]

3.18 With either a nodal heat source or a nodal temperature boundary condition, you have an opportunity to indicate whether you desire nodal heat sources at interior nodes. Enter the nodal heat source value afterwards (Help 3.21). ‘<’ branches back to the selection of the boundary condition type while ‘>’ and ‘Q’ branch further to allow you to decide to enter, delete, or leave boundary condition definition. [124]

3.19 Since you have specified a convection boundary condition, you are asked to specify the surface conductance coefficient h using a consistent set of units. See Segerlind 1984, p 146, eq. 11.35 and p 165, eq. 13.5. [126]

3.20 You need a second value, the fluid temperature, to complete the convection boundary condition (see Help 3.19). [126]

3.21 Enter both nodal temperature and nodal source values in appropriate units here. ‘<’ branches back for the selection of a different boundary condition type (Help 3.15), and ‘>’ branches to the beginning of boundary condition section (Help 3.14). [128]

3.22 Specify boundary conditions at a single node or along the boundary over a range of nodes. Use the left and right arrow keys to move the cursor to the desired starting and ending point; press ‘S’ to select. The program subsequently converts all boundary conditions into nodal equivalents before modifying the global system of equations. Note: You must observe the COUNTERCLOCKWISE convention for the boundary starting and ending points. Should you forget, the program will apply the boundary condition to all of the remaining boundary. To recover, delete the condition or use control-Q to do a warm start to the menu 3.05. Select option 2 and begin entering the boundary conditions again. The program uses the XDRAW command to place the symbols on the screen. This means that if you enter a condition at a node twice, the symbol will disappear and can be misleading. This makes the warm restart approach somewhat safer. [132]

3.23 For multiply-connected regions (i.e., regions having more than one boundary), you are asked whether you wish to specify boundary conditions on another boundary. [134]

3.24 If not in free-run mode, you have an opportunity to have 'second thoughts' and revise any of the boundary conditions. The program saves the boundary condition data (type codes and values) and then calculates and saves the equivalent nodal boundary conditions. [136]

3.25 Supply the name for the data being saved. The number of underscores indicates the maximum name length. Space characters are allowed in the file name. Usually you simply press return to accept the default. The free-run mode does this automatically. Should you wish to place several problems on the diskette by renaming 'FILEINFO.TXT' and preparing another FILEINFO.TXT WITHOUT erasing the diskette, add the problem keyword to the beginning of the file to identify the connection. [180]

3.26 You need to enter the file name of the data you wish to retrieve. Press 'return' to accept the default; free-run mode does this automatically (see Help 3.25). [188]

3.27 If a data file happens to have been locked when you try to save another file with the same name, you have an opportunity to unlock the file here. [208]

3.28 This is the principal PREPROCESS menu (3.05). As usual, progress sequentially through the calculation steps. You must input the properties before applying the boundary conditions because of the way the convection boundary conditions are applied. Options 3-8 allow you to review and check the problem formulation. Remember that tabular data is listed by the screenful. If you need more than one screen, press 'return' for the next screen, press space for the next line, or press a number (0-9) for continuous scroll. Press the space bar to pause. You can change the scroll speed 'on-the-fly'.

Input boundary conditions refer to the user specified conditions, and 'equivalent' refers to the internally determined nodal equivalents of the conditions you specified. [244]

3.29 The exit menu (3.9) presents the standard exit options - proceed to the next program, return to the main menu in the HELLO program, remain in this program, or terminate the calculations. In addition, the program provides a link to a text file creation program. You can customize that program, given only in outline form in this book, to generate an input file for mainframe computer execution. In other words, utilizing Segerlind's programs or the extensions developed by Robert J. Gustafson, you can use the microcomputer with its low-cost graphics to formulate problems that you want to execute on a larger computer (see references).

The exit menu option 3 permits you to use a separate program diskette for the user-supplied program. This approach permits you to write such a program without requiring a detailed knowledge of the tightly compressed PREPROCESS program.

If you do not need the full generality of PREPROCESS (e.g., the element by element variability), then you only need to store integer codes for material property sets, saving enough memory to permit you to formulate much larger problems. [248]

3.30 If you are revising a previously formulated problem, are in demo mode, or are simply changing the size of the problem, answer 'Y' to retrieve the earlier material properties data. [256]

3.31	You have reached this point by completing the input of the material properties data. A 'Y' answer sends you directly to the boundary conditions input; an 'N' returns you to the menu 3.05 (see Help 3.28). [268]
3.32	Retrieve the currently active boundary condition files. If present, these data become the default prompts for the boundary condition re-formulation. If the data are not present or are present but have been designated as inactive in the FILEINFO.TXT file, you are advised of the problem. [276]
3.33	For review, you can list the 'material' properties generated by this program. Recall that you can obtain each new screen of data by pressing 'return'. Press the space bar for a line-by-line display or press '0-9' to set the scrolling speed. You can obtain a printed copy by pressing control-P while the display has paused. [294]
3.34	When you use shading to represent the magnitude of a calculated value, the density of the dots illustrates the range from high to low. If you are using a color monitor, you may need to adjust the shading combinations set in line 230. [304]
3.35	Since you are describing the value, you have that information in both graphical and text formats. [306]

2.3.4 Solve

Help #	Comment [line #]
4.1	This general purpose subroutine allows you to decide when to proceed. Refer to the list of line cross-references to determine the origin of calls. [4]
4.2,4.3	During the extended computations of this program, you can change the level of display details. To reach this point you pressed 'esc'; if you now press 'D'; you can display (1) only the final global matrices, (2) element and global matrices, (3) details of the calculations, or (4) no display. [6,8]
4.4	While applying the convection boundary conditions, you can display (1) the initial global stiffness matrix, (2) the updated global stiffness matrix, (3) both, or (4) neither. [70]
4.5	While applying the constant temperature boundary conditions, you can display (1) the global stiffness matrix already modified for convection, (2) the global stiffness matrix updated for constant temperature boundary conditions, (3) the original (right hand side) source vector, (4) the updated source vector, (5) all of the above, (6) 2 and 4 above, or (7) none of the above. [98]
4.6	You can display the nodal temperatures as you compute them provided you are not in the free-run mode. [124]
4.7	If you are not using the free-run mode, you can select the name for a data file. Normally you accept the default unless you wish to save several versions of the same data file. [154]
4.8	If you are not in free-run, select the specific version of the data file you wish to retrieve. The free-run mode always uses the default. [164]
4.9,4.10	These options allow you to recover from trying to save a data file having the same name as a locked file or to determine the name of a file you wish to retrieve. [188,192]
4.11	This is the principal SOLVE menu (4.05). Choose 1 to proceed through options 2-5 without user interaction. Otherwise, sequentially assemble the global force (i.e., source) and stiffness (i.e., conductance) matrices and update these matrices, first for any convection and then for constant temperature boundary conditions. Finally, solve the system of equations for the nodal temperatures. As is the custom, you can examine the results before leaving this program. [216]

4.12 The exit menu (4.7) routes you to POSTPROCESS, routes you back to the HELLO program, allows you to remain in SOLVE, or allows you to terminate the program. If you only need to plot the mesh, the boundary conditions, and the nodal temperatures, go immediately to PLOT via HELLO. [220]

4.13 Option 1 coordinates the execution of the other calculations. If you have not elected a global free-run while in the HELLO program, you can choose to explore the computational details of the assembly and solution. [230]

2.3.5 Postprocess

Help # Comments [line #]

5.1 This general purpose subroutine allows you to decide when to proceed. Refer to the line cross-reference list to determine the origin of calls to this routine. [4]

5.2 When solving for the element temperatures (using the nodal temperatures), you can elect to display the results as obtained. [20]

5.3 When solving for the element temperature gradients, you can display the components of the gradient, display the resultant (vector sum) of the components in magnitude and angle format, or suppress the display. Option 7 of the principal menu 5.05 (see Help 5.10) presents another opportunity for listing the results. [44]

5.4 Form conjugate matrices and use the Gaussian solution of this new system of equations. You can display the intermediate calculations. [50]

5.5 Option 7 of menu 5.05 allows you to review all the results computed in this program segment. Included are the temperatures (for each node and each element centroid), temperature gradients (at the element centroids or at the nodes), the resultant nodal heat sources, and the surface heat fluxes. Remember that you can use control-P whenever the screen is stopped to obtain a printed output of the results. [70]

5.6,5.7 As is the case for each of the program segments, you can specify the name for a data file. Also, in conformity with the other programs you can specify the file you want to retrieve. Free-run uses the default names. [120,130]

5.8,5.9 To aid you in recovery from disk errors, you can unlock a file or obtain a list of all the data files. [150,154]

5.10 This is the principal POSTPROCESS menu (5.05). Except when studying the computational details, elect option 1 which coordinates the execution of the program parts. Obtain temperatures and temperature gradients at the element centroids and compute nodal temperature gradients and resultant sources and fluxes. As is true throughout the entire package, you can examine the results before proceeding to PLOT (see Help 5.5). [182]

5.11 The exit menu routes you to PLOT, to the MAIN menu of HELLO, back into POSTPROCESS, or out of the program entirely. [184]

5.12 If you did not elect the global free-run in the HELLO program, you are asked whether you wish to examine the progress of the calculational details. [196]

2.3.6 PLOT

Help # Comment [line #]

6.1 This general purpose subroutine allows you to decide when to proceed. Refer to the line cross-reference list for the origin of calls to this routine. [4]

6.2 When presented with graphical results (usually in the mixed graphics mode with four lines of text), you can switch easily from Text (control-T), to mixed graphics (control- G), or to full view graphics (control-V) at any input statement in which the square brackets [] appear. The commands must appear as the first input character. (Note: the single keystroke input obtained through the Applesoft GET statement does not support this control.) [80]

6.3 The highest and lowest temperature values can be identified with the characters 'HI' and 'LO'. (Note: the contour labelling option (Help 6.19) permits the actual contour values or letter codes to be given.) [86]

6.4-6.7 The program presents the minimum and maximum data values for both contour and shading plots. The plot minimum and maximum fall within the range specified here. The default matches the extreme data values. You can select other values to produce more easily read contours or ranges. If you wish to exclude one of the extreme values along a boundary to avoid an "extraneous" line on the plot resulting from the basic nature of the triangular element, see the ideal fluid flow past a circular cylinder in a channel. A triangular element with linear interpolation over the element must necessarily have a constant value throughout if two of the three sides are assigned to the same constant boundary value. Consequently, every pair of two nodes on such a triangle technically falls on the contour. If you are interested in the computed value in the vicinity of such a corner triangle, use the diagonal reversal option in the mesh generation program GRID to force two triangles to share the corner vertex. The graphics resolution restricts you to eleven contour lines. (Note: Any time your input is not accepted, you can use control-R to examine any restrictions. [102,104]

6.8 If you are working with a complicated mesh and wish to increase the resolution of your hi-res graphics, subdivide the screen into equal-sized rectangles and draw each of these subdivisions to fill the screen. Join screen dumps of the results for a more detailed view. Corner markers facilitate the re-combination process. [118]

6.9 In the upper-right corner of the screen is a map of the regions selected for enlargement. You can plot all zones, selected zones, or none. You select by designating those to be included or those to be excluded. This process overcomes the coarseness of the graphics at the expense of computational time. The entire figure is re-generated for each zone with the unneeded lines clipped (see Help 6.10 also). [120]

6.10 The zone plotting options provide great flexibility in getting increased resolution. If you want to enlarge a specific portion of the screen, select an arbitrary rectangular region to fill the entire screen. To avoid distortion, this is subject to the maintenance of the width to height (aspect) ratio. Alternatively, you can generate multiple equal-sized zones (see Help 6.8 and 6.9). [150]

6.11 Redo the last zone plot if you are not satisfied. [154]

6.12 If you have made zone plots, you can re-select zones for further exploration. [158]

6.13 If you do not wish to save the plot or to take the time to label the plot, bypass the questions about labels for elements, nodes, contours, or captions. [162]

6.14,6.15 If you wish to add labels to elements or nodes, either automatically label all or selectively label those of particular interest. To distinguish element and node numbers, an 'E' precedes element numbers. (see Help 6.16 and 6.17). [164,166]

6.16,6.17 Use left and right arrows to move the cursor from one label to the next. When pointing to one you wish to edit, press 'S' to select it. Use 'X' to add or remove the label (using the XDRAW command). To move the label in 'large' increments, use the traditional I,J,K,M diamond for up, left, right, or down movement. If you press the control key at the same time, control-I, control-J, control-K, and control-M move the cursor in smaller steps (1/10). When you have positioned the label correctly, press 'Q'. Select another label or press 'Q' a second time to quit editing. [174,176]

6.18 Since multiple lines and labels clutter a figure, you may choose to redraw the label. This makes the labels more prominent. Since labels are initially placed on the screen with the XDRAW command, those which overlap lines or other labels may become partly obscured. The redraw option first erases the screen where the label is currently placed then redraws the label. Unless over-written by another label, the legibility is now enhanced; however, you can no longer move the label. [176]

6.19-6.21 Identify a contour line by value or by a letter associated with a value in the legend (printed later on the text screen). If you wish to label a contour, use the left and right arrows to move the cursor from contour to contour. Press 'S' to select a contour to be labeled; then specify the actual location of the label using the I, J, K, and M diamond to move the label. Use 'X' to add or delete the label. 'Q' allows you to quit. [180,182]

6.22 When letters identify the contour lines, examine the legend on the text page by pressing control-T. Use control-V to return to the full graphics view. Then press 'return' to continue. [184]

6.23 See Help 6.18 for a discussion of the redraw option. [186]

6.24-6.27 Now you can add text labels to the plot. See 6.19-6.21 for the commands. The only difference in the implementation is that you must provide the text of the label and initially locate your first guess for the center of the label. Note that you can elect to remove the label using 'X' and redraw later to enhance legibility. [188,190,192]

6.28 After generating a plot, you must specify whether to save the picture on the data diskette, obtain an enlargement, or simply abandon this picture.

Each picture requires 33 diskette sectors (of 256 bytes each) to save. If storing more than 8 to 10 figures (usually not necessary), you may need an additional data diskette. If you have a screen dump program resident in the $1D00.1FFF portion of memory, you can also make a screen dump by simply pressing control-P. [196]

6.29 If you save the picture, the keyword will be a prefix to the file name selected and '.PIC' will be appended to identify this as a picture file. See Help 6.38 for a discussion of the use of the suffix. [198]

6.30 This requests the name of the data file to be loaded. The default is usually sufficient. The free-run mode uses the default. [228]

6.31,6.32 Use disk error recovery to unlock a file or to review the entire list of data file names. [246,250]

6.33 This is the principal menu (6.5) and provides an option for plotting each of the classes of calculated results. (Each of the earlier programs provided tabular listings; only plots are available in PLOT.)

(1) Plot the entire generated grid. See the earlier help comments about element, node, contour, and text labels, as well as zone plots.

(2) Plot the boundary outlines and indicate the boundary condition type. A boundary with no symbol added signifies a zero-flux boundary condition. Circles at nodes indicate a prescribed temperature. Four arrows pointing outward denote a heat source while arrows pointing inward denote a heat sink. A dashed line denotes convection.

(3) Plot nodal temperatures as contours. Usually contours are plotted with a solid line for the boundary. If needed, you may superimpose the generating mesh on the plot using dashed lines. Such a plot can easily become cluttered but can clarify the interpolation process. The various zone and label options are available.

(4) Represent element temperatures by shading each element according to the magnitude of the element temperature.

(5) & (6) Represent nodal and element temperature gradients either with contour plots, shading, or arrows which represent the magnitude and direction of the vector.

(7) Finally, retrieve previously saved pictures for viewing or for plotting if you have a screen dump capability. [284]

6.34 The exit menu has the usual options. This is frequently the program termination point. [286]

6.35 This specifies the gradient component to be plotted for each coordinate direction or for the resultant vector. [294]

6.36 Select the type of plot that best represents the gradient information. For nodal temperature gradients, choose either arrows to represent the magnitude and direction (which is quick but has limited resolution) or contour lines (which provide greater resolution but take longer). If you are plotting element temperature gradients, your choices are shading and arrows. Both can be completed rather quickly. [298]

6.37 Superimpose the plot on the boundary outline of the problem. Option 2 permits you to use a dashed mesh as background for the plot. This option can be effective as a diagnostic technique but often leads to figures which are 'too busy'. [302]

6.38,6.39 When you wish to retrieve a picture from the diskette, you have a list of file names which end with '.PIC'. Select one by entering only the menu item number, not the entire file name. This also displays the number of free sectors on the diskette. The picture is displayed in mixed text and graphics mode. Use control-T or control-V to change the display. Control-G returns you to the present display format.

Once you retrieve the results, you can review them or print them if you have a 'Grappler' screen dump card. You can modify Line 320 to handle other screen dump programs or cards. [314,318]

2.3.7 Diskette Preparation

Help # Comment [line #]

7.1 Remember to remove your program diskette. To avoid accidental erasure, you should always write-protect (tab) the program diskette. [4]

7.2 Since you will be asked whether to erase the diskette, this is one last opportunity for you to review the names of the files. If this is not a formatted diskette, you will hear some unpleasant sounds; but the program will recover. [6]

7.3 You can abort the effort by deciding not to continue. In that case you are presented with the exit menu next. Otherwise, proceed with the diskette preparation.

When asked if you want to erase (re-format) the diskette, press 'Y' or 'N' and 'return'. 'Y' reformats the diskette as a data diskette. Neither a greeting program nor the Disk Operating System is on the diskette. Only sector 0 of track 0 and track $11 are unavailable for data. Such a diskette does not boot-up the system. 'N' preserves the current contents of the diskette. If you remember to rename the 'FILEINFO.TXT' file first, you can place more than one problem on the diskette. Otherwise, the next step erases this vital information, rendering the existing problem unusable. [6]

7.4 To identify all of the related data files, use the keyword requested here. This short keyword precedes each data file name. The remainder of the file name simply indicates the nature of the data in the file. Choose a keyword which will help you identify the problem. [8]

7.5 The problem description allows you more flexibility in defining the problem. You can use up to 80 characters. You can use spaces freely; however, you CANNOT use a comma or a colon. By pressing 'return' you create the problem directory file called 'FILEINFO.TXT' erasing any previous file by that name. [8]

7.6,7.7 You can review the contents of the file created in this step. You can also review the contents of other random access and sequential text files in direct mode (i.e., when no program is executing) by issuing the command 'TDUMP file name'. Use control-S to stop and start the display. Use control-C to exit the file dump prematurely. [30]

7.8 The exit menu (7.10) permits you to return to the MAIN MENU in HELLO, to prepare another diskette, or to terminate the program. [32]

7.9 The program diskette must be in drive 1. Only the single-drive users must swap diskettes here. [34]

7.10 This general subroutine awaits a user response to continue. [46]

2.4Annotated Source Listings for Heat Conduction

2.4.0 Source Listings

HELLO.HEAT

At boot up, load the language card with DOS and part of the utilities.

```
2  TEXT :
   HOME :
   IF  PEEK (978) = 157 THEN
       PRINT "[M]LOADING THE LANGUAGE CARD[M][M][M]
          ONE MOMENT, PLEASE...[M][M][M]UTILITIES: PART 1":
       PRINT  CHR$ (4)"BLOAD FEM-B,A$2F70":
       CALL 12144:
       PRINT "[M][M]MOVING DOS TO BANK 2":
       PRINT  CHR$ (4)"BLOAD FEM-A,A$4000":
       CALL 16384:
       REM HELLO(H)
```

Place the 'Hello' program above hi-res page 1.

```
4  VTAB 12:
   PRINT "LOCATE APPLESOFT ABOVE HI-RES PAGE 1":
   PRINT  CHR$ (4)"BLOAD LOADER":
   CALL 768:
   D$ =  CHR$ (4):
   WR$ = "... WHEN READY, PRESS <RETURN>":
   SR$ = "... SELECT BY NUMBER <RETURN>":
   ON  PEEK (2051) = 169 GOTO 8
```

If not already loaded, put additional utilities below hi-res page 1.

```
6  PRINT "[M][M]LOADING UTILITIES (PART 2)
      AND SYMBOLS":
   PRINT D$"BLOAD FEM-C,A$803,D1":
   CALL 2051:
   & D(T):
   POKE 2165,1
```

Identify the program and if necessary, reload the shape table and hi-res dump.

```
8  ON  PEEK (6462) = 75 AND  PEEK (6463)
      = 75 GOTO 10:
   PRINT D$"BLOAD SHAPES,A$193E,D1":
   PRINT D$"BLOAD HI-RES DUMP,A$1D00,D1"
```

If using a II-e or II-c, set 'caps lock'.

```
10  POKE 2164,0:
    HOME :
    POKE 232,62:
    POKE 233,25:
    IF  PEEK (64435) = 6 THEN
        PRINT "[M][M]SET CAPS LOCK AND":
      & I(WR$;"";A$,"1")
```

Set the number of disk drives. If a dual drive, insert the data diskette.

```
12  A =  PEEK (2048):
    ON A = 1 OR A = 2 GOTO 14:
    HOME :
    & B(20,20):
    & I("[M][M]HOW MANY DISK DRIVES ARE YOU USING";
```

```
      2;A,"2",0,0,A = 1 OR A = 2):
   POKE 2048,A:
   ON A = 1 GOTO 14:
   & B(200,20):
   PRINT "[M][M]INSERT ";:
   FLASH :
   PRINT "DATA";:
   NORMAL :
   PRINT " DISK INTO DRIVE 2[M][M]":
   & I(WR$;"";A$,"3")
```

(14-16) Set the level of user interaction.

```
14 HOME :
   PRINT "* USER INTERACTION MODE *[M][M]
      1. MAXIMAL (NO DEFAULTS)[M]
      2. INTERMEDIATE (SOME DEFAULTS)[M]
      3. MINIMAL (FREE RUN)[M]
      4. DEMO WITH PREVIOUS DATA[M]":
   & I(SR$,2;A,"4",BC%,1,A >0 AND A <5):
   ON BC% GOTO 12:
   POKE 2163,A - 1

16 IF A = 4 THEN
      & I("[M][G]WARNING: CALCULATED DATA WILL NOT BE[M]
         SAVED! OK? (Y/N)";"N";A$,"5",0,1):
      ON A$ < >"Y" GOTO 14
```

Main title.

```
18 HOME :
   INVERSE :
   FOR I = 1 TO 4:
      PRINT  TAB( 2)" " TAB( 38)" ":
   NEXT I:
   NORMAL :
   VTAB 2:
   HTAB 3:
   PRINT " APPLIED FINITE ELEMENT ANALYSIS: ":
   HTAB 3:
   PRINT "     AN APPLE " CHR$ (221) CHR$ (219);"
   " IMPLEMENTATION    ":
   VTAB 6

20 HTAB 13:
   INVERSE :
   PRINT "HEAT CONDUCTION[M][M]":
   NORMAL :
   PRINT  TAB( 19)"BY[M][M]" TAB( 7)
      "J. R. COOKE AND D. C. DAVIS[M][M][M][M]"
      TAB( 13) "COPYRIGHT, 1985[M][M]" TAB( 8 )
      "JOHN WILEY AND SONS, INC.[M][M]" TAB( 11)
      "ALL RIGHTS RESERVED":
   VTAB 23:
   & I(WR$;"";A$,"6",BC%):
   ON BC% < >0 GOTO 10

22 DATA "GEOMETRY","DEFINE PROBLEM GEOMETRY",3,"GRID",
      "GENERATE & MODIFY GRID",5,"PREPROCESS",
      "DEFINE PROPERTIES AND BOUNDARY        CONDITIONS",7,
      "SOLVE",
      "ASSEMBLE EQUATIONS & SOLVE FOR NODE  TEMPERATURES",10,
      "POSTPROCESS"

24 DATA "SOLVE FOR AVERAGE TEMPERATURES AND GRADIENTS",
      13,"PLOT",
```

```
          "PLOT GRID, BOUNDARY CONDITIONS, &     PROBLEM RESULTS",
          16,"DISKETTE PREPARATION","PREPARE DATA DISKETTE",19:
       RESTORE :
       FOR I = 1 TO 7:
           READ PN$(I),PD$(I),LI(I):
       NEXT I

              The 'MAIN MENU'.
26     N =  PEEK (2165):
       HOME :
       PRINT "***     MAIN PROGRAM MENU (0.00)      ***[M]":
       FOR I = 1 TO 7:
           PRINT I". ";:
           INVERSE :
           PRINT PN$(I):
           NORMAL :
           PRINT "    " LEFT$ (PD$(I),36);:
           IF  LEN (PD$(I)) >36 THEN PRINT "    "
              RIGHT$ (PD$(I), LEN (PD$(I)) - 36);

28         PRINT :
       NEXT I:
       PRINT "[M]0. STOP[M]":
       POKE 34,22:
       & I(SR$;N;N,"7",0,1,N > = 0 AND N <8):
       POKE 34,0:
       IF N = 0 THEN
            END

30     IF  PEEK (2163) = 2 THEN
            & B(10,10):
           & B(0,0,10)

              (32-36) Activate the selected program.
32     VTAB LI(N):
       INVERSE :
       PRINT N". "PN$(N) TAB( 40)" ":
       POKE 2165,N:
       POKE 2166,1:
       NORMAL :
       IF N = 4 OR N = 5 THEN
            POKE 103,63:
           POKE 104,25:
           POKE 6462,0:
           REM     $193E

34     ONERR  GOTO 38

36     PRINT D$:
       PRINT D$"RUN "PN$(N);DR$(DR)

              Error handling.
38     POKE 34,22:
       VTAB 23:
       POKE 216,0:
       HOME :
       PRINT  CHR$ (7) CHR$ (7):
       & I("INSERT CORRECT PROGRAM DISK <RETURN>";
          "";A$,"8"):
       GOTO 36

65535 REM [M][M]1FEB85[M]JRC/DCD

END-OF-LISTING
```

GEOMETRY.HEAT

Initialize; branch to main menu.

```
1  TEXT :
   & D(T):
   & L(255):
   P% = 10:
   Z =  PEEK (2163):
   POKE 2164,1:
   D$ =  CHR$ (4):
   WR$ = "... WHEN READY, PRESS <RETURN>":
   SR$ = "... SELECT BY NUMBER <RETURN>":
   R$ = "[M][M]RETAIN HOW MANY ":
   FI$ = "FILEINFO.TXT":
   GOTO 264:
   REM GEO(H/E)
```

Warm restart entry.

```
2  GOTO 282
```

Wait for user response, clear screen and continue.

```
4  PRINT :
   & L(255):
   VTAB 24:
   CALL  - 868:
   & B(20,20):
   & I(WR$;"";A$,"1",BC%):
   HOME :
   RETURN
```

Free-run beep.

```
6  ON Z < >2 GOTO 4:
   & B(10,10):
   & B(0,0,8):
   & L(255):
   HOME :
   RETURN
```

(8-10) Cursor control for deleting a point.

```
8  HOME :
   PRINT "DELETE-[M]  MOVE CURSOR USING ARROWS":
   POKE 34,22:
   N = (N = 0) +  N * (N < >0)

10 SCALE= 1:
   XDRAW 67 AT  FN PX(X(N)), FN PY(Y(N)):
   HOME :
   PRINT "  ( D Q LEFT ARROW  RIGHT ARROW )":
   GET A$:
   A =  ASC (A$):
   XDRAW 67 AT  FN PX(X(N)), FN PY(Y(N)):
   N = N +  FN S(NP):
   ON A$ = "Q" GOTO 108:
   ON A$ < >"D" OR NP = 0 GOTO 10:
   S = 2:
   SY = 64:
   GOSUB 34:
   GOSUB 20:
   GOTO 10
```

Does point already exist?

```
12  OL = 0:
       ON NP = 0 GOTO 58:
       FOR I = 1 TO NP:
          OL = OL +  FN EQ(NO):
          I = I +  OL * NP:
       NEXT I:
       RETURN
```

Draw square symbol on region side for nodes per side request.

```
14  FOR I = 1 TO 11:
       XDRAW 65 AT  FN PX(X(GS%(I))), FN PY(Y(GS%(I))):
       I = I +  11 * (GS%(I +  1) = 0):
    NEXT I:
    RETURN
```

Find location and set value for nodes per side of region.

```
16  FOR I = 2 TO 8 STEP 2:
       I1 = I / 2:
       I = I +  8 * (GN%(RS,I) = GS%(1)):
    NEXT I:
    ON (NC%(RS) * (I1 = 1 OR I1 = 3) +  NR%(RS) * (I1
      = 2 OR I1 = 4) < >0) GOSUB 18:
    RETURN
```

Label sides with number of nodes specified.

```
18  FOR I = 1 TO 11:
       XDRAW NC%(RS) * (I1 = 1 OR I1 = 3) +  NR%(RS)
          * (I1 = 2 OR I1 = 4) +  17 AT  FN PX(X(GS%(I)))
          - 8, FN PY(Y(GS%(I))) +  9:
       I = I +  11 * (GS%(I +  1) = 0):
    NEXT I:
    RETURN
```

(20-22) Delete a previous point.

```
20  ON N = NP GOTO 22:
    FOR I = N TO NP - 1:
       X(I) = X(I +  1):
       Y(I) = Y(I +  1):
    NEXT I:
    X(NP) = 0:
    Y(NP) = 0

22  NP = NP - 1:
    N = N * (N <NP) +  NP * (N > = NP):
    RETURN
```

Check region points for duplicates or inappropriate vertex or midpoint of another region.

```
24  FOR I = 1 TO 8:
       O1 = (GN%(RE,I) = N):
       I = I +  8 * O1:
    NEXT I:
    EV = (2 *  INT (ND / 2) = ND):
    FOR I = 1 TO RE:
       FOR J = 2 - EV TO 8 STEP 2:
          O2 = (GN%(I,J) = N):
          J = J +  8 * O2:
       NEXT J:
       I = I +  10 * O2:
    NEXT I:
    A1 = (O1 +  O2 >0):
    RETURN
```

Delete a region point.

```
26  LN = ND:
    ND = ND - (ND >1):
    ON LN - ND = 0 GOTO 58:
    N = GN%(RE,ND):
    XDRAW 65 AT  FN PX(X(N)), FN PY(Y(N)):
    GN%(RE,ND) = 0:
    RETURN
```

Plot a region boundary and mark the 8 points with '+'.

```
28  ON GN%(RE,0) = 0 GOTO 58:
    HCOLOR= 3 * DF:
    N1 = GN%(RE,1):
    HPLOT  FN PX(X(N1)), FN PY(Y(N1)):
    FOR I = 1 TO 8:
        N1 = GN%(RE,I +  1 - 8 * (I = 8)):
        HPLOT  TO  FN PX(X(N1)), FN PY(Y(N1)):
        & B(10,10):
        & B(0,0,5):
    NEXT I:
    ON DF GOTO 58
```

Delete a region.

```
30  SCALE= 2:
    FOR I = 1 TO 8:
        HCOLOR= 3:
        N1 = GN%(RE,I):
        DRAW 64 AT  FN PX(X(N1)), FN PY(Y(N1)):
        HCOLOR= 0:
        HPLOT  FN PX(X(N1)), FN PY(Y(N1)):
    NEXT I:
    HCOLOR= 3:
    SCALE= 1:
    RETURN
```

Place region number at its centroid.

```
32  ON GN%(RE,0) = 0 GOTO 58:
    XC = 0:
    YC = 0:
    FOR I = 1 TO 8:
        XC = XC +  X(GN%(RE,I)):
        YC = YC +  Y(GN%(RE,I)):
    NEXT I:
    XC = XC / 8:
    YC = YC / 8:
    L$ =  STR$ (RE):
    FOR I = 1 TO  LEN (L$):
        XDRAW  ASC ( MID$ (L$,I,1))
            - 31 AT  FN PX(XC) - 7 *  LEN (L$) +  7
            * (I - 1) +  4, FN PY(YC) +  3:
    NEXT I:
    RETURN
```

Place symbol number SY at scaled location.

```
34  SCALE= S:
    XDRAW SY AT  FN PX(X(N)), FN PY(Y(N)):
    RETURN
```

Draw symbols at all 8 points of region.

```
36  S = 1:
    SY = 65:
    FOR J = 1 TO 8:
        N = GN%(RE,J):
        ON N < >0 GOSUB 34:
```

```
    NEXT J:
    RETURN
```

Define region points in counter-clockwise order.

```
38  HOME :
    PRINT "DEFINE POINTS (CCW) FOR REGION "RE:
    N = (N = 0) +  N * (N < >0):
    ND = 1:
    FOR I = 0 TO 8:
        GN%(RE,I) = 0:
    NEXT I:
    PRINT "USE ARROWS TO MOVE CURSOR":
    POKE 34,22:
    RETURN
```

Specify a region to be deleted.

```
40  HOME :
    PRINT "DELETE A REGION-":
    & I("  REGION NUMBER=";RE;RE,"2",
       BC%,0,RE >0 AND RE < = NQ):
    ON BC% >0 GOTO 58:
    RS = RE:
    GOSUB 220:
    DF = 0:
    RE = RS:
    GOSUB 28:
    FOR K = RS TO NQ:
        FOR J = 1 TO 8:
            GN%(K,J) = GN%(K +  1,J):
    NEXT J,K:
    GN%(NQ,0) = 0:
    NQ = NQ - 1:
    GOSUB 220:
    RE = RS - (RS >1):
    RETURN
```

(42-48 and 52-58) Locate region sides, which must have the same number of nodes.

```
42  & C(GS%,CN%):
    DIM GS%(11),CN%(10):
    RS = RE:
    N0 = N:
    N1 = N +   FN WN(N):
    GS%(1) = GN%(RE,N * 2):
    GS%(2) = GN%(RE,N1 * 2):
    NS%(1) = N:
    NS%(2) = N1:
    GI = 3:
    CN%(RE) = N:
    HOME :
    PO = 14:
    PRINT "REGION: "RE"[M]SIDES:  "N" "N1

44  K = 1:
    TN = N:
    ON (JT%(RE,N * (N < >0) +  (N = 0))
       < >0) AND (N < >0) GOTO 46:
    K = 2:
    TN = N1:
    A = ((JT%(RE,N1 * (N1 < >0) +  (N1 = 0))
       = 0) OR (N1 = 0)):
    ON A GOTO 52

46  NS%(K) = 0:
    LN = GN%(RE,TN * 2):
```

```
    RE = JT%(RE,TN):
    ON RE = RS GOTO 52:
    FOR I = 1 TO 4:
       IF GN%(RE,2 * I) = LN THEN
          N = I:
          I = 5

48  NEXT I:
    CN%(RE) = N:
    N1 = N + FN WN(N):
    VTAB 22:
    HTAB PO:
    PRINT RE:
    HTAB PO:
    PRINT N" "N1:
    PO = PO + 5:
    N = N * (GN%(RE,N1 * 2) = LN) + N1 * (GN%(RE,N
      * 2) = LN):
    GS%(GI) = GN%(RE,N * 2):
    GI = GI + 1:
    TN = N:
    ON JT%(RE,N) < >0 GOTO 46:
    RE = RS:
    N = NS%(1):
    N1 = NS%(2):
    GOTO 44
```

Draw a square at scaled XC,YC.

```
50  ON XC <UL OR XC >UH OR YC
      >VH OR YC <VL GOTO 58:
    SCALE= 1:
    XDRAW 65 AT FN PX(XC), FN PY(YC):
    RETURN
```

(See line 42).

```
52  GI = GI - (A = 0):
    GS%(GI) = (A = 1) * GS%(GI):
    GOSUB 14:
    A = 3:
    ON ED = 0 GOTO 54:
    A = RD%(RS,6):
    ON NS%(1) = 1 OR NS%(1) = 3 GOTO 54:
    A = RD%(RS,5)

54  VTAB 24:
    & I("ENTER NUMBER OF NODES (2-9)";A;A,"3",BC%,0,A
      >1 AND A <10):
    ON BC% < >0 GOTO 58:
    GOSUB 14:
    GOSUB 16

56  FOR I = 1 TO NQ:
       NC%(I) = NC%(I) * (CN%(I) = 0 OR CN%(I)
          = 2 OR CN%(I) = 4) + A * (CN%(I)
          = 1 OR CN%(I) = 3):
       NR%(I) = NR%(I) * (CN%(I) = 0 OR CN%(I)
          = 1 OR CN%(I) = 3) + A * (CN%(I)
          = 2 OR CN%(I) = 4):
    NEXT I:
    I1 = N0:
    GOSUB 18:
    RE = RS:
    N = N0
58  RETURN
```

(60-64) Define coordinates of a single point.

```
60  POKE 34,20:
    HOME :
    PRINT "ADD A SINGLE POINT-":
    POKE 34,21:
    & I(DF$(1) +  " (" +   STR$ (UL) +  "-" +   STR$ (UH)
       +  " ) =";X(NP +  1);X(NP +  1),"4",BC%,0,X(NP
       +  1) > = UL AND X(NP +  1) < = UH):
    ON BC% < >0 GOTO 108

62  & I(DF$(2) +  " (" +   STR$ (VL) +  "-" +   STR$ (VH)
       +  ") =";Y(NP +  1);Y(NP +  1),"5",BC%,0,Y(NP
       +  1) > = VL AND Y(NP +  1) < = VH):
    ON BC% GOTO 60,108,108

64  S = 2:
    SY = 64:
    NO = NP +  1:
    GOSUB 12:
    O1 = OL:
    NP = NP +  (O1 = 0):
    N = NP:
    ON O1 = 0 GOSUB 34:
    ON O1 = 0 GOTO 108:
    INVERSE :
    PRINT "POINT ALREADY DEFINED":
    NORMAL :
    GOSUB 4:
    GOTO 60
```

Select a previously defined point.

```
66  HOME :
    PRINT "ARROWS MOVE 'X' CURSOR[M]<S>SELECTS":
    N = (N = 0) +  N * (N >0 AND N < = NP)
       +  NP * (N >NP):
    X1 =  FN PX(X(N)):
    Y1 =  FN PY(Y(N)):
    SCALE= 1:
    XDRAW 67 AT X1,Y1:
    GET A$:
    A =  ASC (A$):
    XDRAW 67 AT X1,Y1:
    N = N +  FN S(NP):
    ON A$ < >"S" GOTO 66:
    RETURN
```

Define coordinates of a point on a straight line.

```
68  X(N) = X(N1) +  FR * (X(N2) - X(N1)):
    Y(N) = Y(N1) +  FR * (Y(N2) - Y(N1)):
    RETURN
```

Define coordinates of a point on a circular arc.

```
70  A0 = A1 +  FR * AN * RD:
    X(N) = XC +  RA *  COS (A0):
    Y(N) = YC +  RA *  SIN (A0):
    RETURN
```

(72-76) Define endpoint for straight line.

```
72  PRINT "- DEFINE ENDING POINT"UP$(NP >0):
    & I("  " +  DF$(1) +  "=";X(N1);X(N),"6",
       BC%,0,X(N) > = UL AND X(N) < = UH):
    ON BC% = 3 AND NP = 0 GOTO 92:
    ON BC% GOTO 92,92,76
```

```
74  & I("  " +  DF$(2) +  "=";Y(N1);Y(N),"7",
      BC%,0,Y(N) > = VL AND Y(N) < = VH):
    ON BC% GOTO 92,74,92

76  ON BC% = 3 GOSUB 66:
    X(N2) = X(N):
    Y(N2) = Y(N):
    N = N2:
    A0 = (X(N1) = X(N2) AND Y(N1) = Y(N2)):
    & B(50 * A0,50 * A0):
    ON A0 GOTO 92:
    S = 3:
    SY = 64:
    GOSUB 34:
    S = 2:
    O2 = (BC% = 3):
    ON BC% = 3 GOTO 96:
    NO = N:
    GOSUB 12:
    O2 = OL:
    ON O2 = 0 GOSUB 34:
    GOTO 96
```

Save endpoint coordinates.

```
78  N2 = N1 +  NI +  1:
    X(N2) = X(N1 +  1):
    Y(N2) = Y(N1 +  1):
    N = N1:
    RETURN
```

(80-102) Define coordinates of multiple points spaced on a straight line or circular arc.

```
80  POKE 34,20:
    HOME :
    PRINT "DEFINE A "LD$(LD)"-":
    POKE 34,21:
    ON LD = 2 GOTO 86:
    PRINT "-DEFINE CENTER OF CIRCLE"UP$(NP >0):
    POKE 34,21:
    & I("  " +  DF$(1) +  "=";XC;XC,"8",BC%):
    ON BC% = 3 AND NP = 0 GOTO 80:
    ON BC% GOTO 108,108,84

82  & I("  " +  DF$(2) +  "=";YC;YC,"9",BC%):
    ON BC% GOTO 80,108,80

84  ON BC% = 3 GOSUB 66:
    XC = XC * (BC% < >3) +  X(N) * (BC% = 3):
    YC = YC * (BC% < >3) +  Y(N) * (BC% = 3):
    GOSUB 50

86  HOME :
    N = NP +  1:
    N1 = N:
    PRINT "DEFINE STARTING PT"UP$(NP >0):
    & I("  " +  DF$(1) +  "=";X(NP);X(N),"10",
      BC%,0,X(N) > = UL AND X(N) < = UH):
    ON BC% = 3 AND NP = 0 GOTO 86:
    ON BC% = 3 GOTO 90:
    ON BC% >0 AND LD = 3 GOSUB 50:
    ON BC% GOTO 80,108

88  & I("  " +  DF$(2) +  "=";Y(NP);Y(N),"11",
      BC%,0,Y(N) > = VL AND Y(N) < = VH):
```

```
    ON BC% = 1 OR BC% = 3 GOTO 86:
    ON BC% = 2 AND LD = 3 GOSUB 50:
    ON BC% = 2 GOTO 108

90  ON BC% = 3 GOSUB 66:
    X(N1) = X(N):
    Y(N1) = Y(N):
    N = N1:
    S = 3:
    SY = 64:
    GOSUB 34:
    S = 2:
    O1 = (BC% = 3):
    IF BC% < >3 THEN
        NO = N:
        GOSUB 12:
        O1 = OL:
        ON O1 = 0 GOSUB 34

92  HOME :
    N = N1 +  1:
    N2 = N:
    ON LD = 2 GOTO 72:
    & I("DEFINE TOTAL ANGLE OF ARC[M] DEGREES (CCW+ ) =";
      AN;AN,"12",BC%,0,AN < >0):
    ON BC% < >0 GOTO 92:
    A1 =  FN A(N1):
    AR = A1 + AN * RD:
    RA =  SQR ((X(N1) - XC) ^ 2 + (Y(N1) - YC) ^ 2):
    X(N) = XC +  RA *  COS (AR):
    Y(N) = YC +  RA *  SIN (AR)

94  A0 = (X(N) <Z1 OR X(N) >Z2 OR
      Y(N) <Z3 OR Y(N) >Z4 OR RA = 0):
    & B(50 * A0,50 * A0):
    ON A0 GOTO 92:
    NO = N:
    GOSUB 12:
    O2 = OL:
    I = N1:
    O2 = (O2 +  FN EQ(N2) >0):
    S = 3:
    SY = 64:
    GOSUB 34:
    ON  FN EQ(N2) GOSUB 34:
    S = 2:
    ON O2 = 0 GOSUB 34

96  HOME :
    NI = 0:
    & I("# OF INTERMEDIATE POINTS";1;NI,"13",
      BC%,0,NI > = 0):
    ON BC% GOTO 92,102,96:
    ON NI = 0 GOTO 102:
    & I("EVENLY SPACED";"Y";A$,"14",BC%,1):
    ON BC% GOTO 96,102,96:
    ON O2 = 0 GOSUB 78:
    ON A$ = "N" GOTO 100

98  S = 2:
    SY = 64:
    FOR I = 1 TO NI:
        FR = I / (NI +  1):
        N = N +  1:
```

```
        ON LD - 1 GOSUB 68,70:
        GOSUB 34:
    NEXT I:
    GOTO 102

100 PRINT "DEFINE DISTANCE (0 TO 1) FROM START":
    S = 2:
    SY = 64:
    FOR I = 1 TO NI:
        & I("- FOR POINT #" +  STR$ (I) +  ": ";I / (NI
           +  1);FR,"15",BC%,0,FR >0 AND FR <1):
        N = N +  (BC% = 0):
        ON LD - 1 GOSUB 68,70:
        ON BC% = 0 GOSUB 34:
        I = I - (BC% < >0):
    NEXT I

102 S = 3:
    SY = 64:
    N = N1:
    GOSUB 34:
    N = N2:
    ON LD = 2 GOSUB 34:
    I = N1:
    ON  FN EQ(N2) = 0 AND LD = 3 GOSUB 34:
    ON LD = 3 GOSUB 50:
    NP = NP +  NI +  2:
    N = N2:
    ON O2 GOSUB 20:
    N = N1:
    ON O1 GOSUB 20:
    GOTO 108
```

(104-112) Define points needed for region definition.

```
104 TEXT :
    & D(T):
    HOME :
    PRINT "GEOMETRY: COORDINATES (1.31)[M][M]
       INSTRUCTIONS:[M][M]ENTER POINTS TO BE USED TO
       DEFINE 8-NODEMESH GENERATING REGIONS WITH 4 VERTICES
       & A NODE ON EACH OF THE 4 (STRAIGHT OR  CURVED) SIDES.
       A TRIANGULAR REGION MAY"

106 PRINT "BE CREATED IF TWO CONNECTED STRAIGHT   LINES
       FORM ONE SIDE.  ADJACENT REGIONS MUST SHARE 3
       POINTS ON THE COMMON SIDE.":
    GOSUB 6:
    ON BC% >0 GOTO 58:
    GOSUB 206:
    GOSUB 218:
    ON BC% GOTO 58,114,104:
    SI%(1) = 0:
    VTAB 21

108 POKE 34,20:
    HOME :
    PRINT "INPUT OF POINTS-[M]
       OPTIONS: ENTER, DELETE, QUIT, BRANCH":
       & I("( E D Q <>)";"E";
       A$,"16",BC%,0,A$ = "D" OR A$ = "E" OR A$ = "Q"):
    HOME :
    ON BC% GOTO 58,154,108:
    AC = (A$ = "E") +  2 * (A$ = "D") +  3 * (A$ = "Q"):
    ON AC GOTO 110,8,154
```

```
110  HOME :
     PRINT "ENTER-[M]
        OPTIONS: POINT, LINE, ARC, QUIT, BRANCH":
     & I("  ( P L A Q <>)";"L";A$,"17",
        BC%,0,A$ = "P" OR A$ = "L" OR A$ = "A" OR A$ = "Q"):
     ON BC% GOTO 108,108,110:
     LD = (A$ = "P") +  2 * (A$ = "L") +  3 * (A$ = "A"):
     ON LD GOTO 60,86,80:
     GOTO 108

112  GOSUB 180:
     POKE 2166,MO +  2:
     RETURN
```

(114-130) Select 8 points in counter-clockwise order to define a region.

```
114  ON SI%(1) = 0 GOSUB 192:
     TEXT :
     & D(T):
     HOME :
     PRINT "GEOMETRY: REGION DEFN'S (1.32)[M][M]
        INSTRUCTIONS:[M][M]BEGINNING WITH A VERTEX,
        DEFINE EACH     REGION WITH A ";:
     INVERSE :
     PRINT "COUNTER-CLOCKWISE";:
     NORMAL :
     PRINT " SEQUENCEOF 8 POINTS."

116  PRINT "[M]USE ARROWS TO POSITION THE CURSOR. THEN[M]
        PRESS <S>TO SELECT.[M][M]
        EACH REGION IS NUMBERED AT ITS CENTROID.":
     DF = 1:
     GOSUB 6:
     ON BC% GOTO 58,132,114:
     RE = 1:
     VTAB 21:
     & D(G):
     IF MO = 3 THEN
          GOSUB 206:
         GOSUB 218:
         IF ED = 1 THEN
              GOSUB 220:
             GOTO 128
```

(118-122) Specify region number and 8 points.

```
118  POKE 34,20:
     HOME :
     PRINT "ENTER A NEW REGION-":
     & I("  REGION NUMBER=";NQ +  1;RE,"18",BC%,0,RE
        = 0 OR RE = NQ +  1):
     ON BC% GOTO 114,128,118:
     ON RE = 0 GOTO 128:
     SI%(0) = 1:
     GOSUB 38:
     DIM B$(1),C$(1):
     B$(0) = "  ":
     B$(1) = "MARKING POINT ":
     C$(0) = "8 POINTS MARKED"

120  HOME :
     A1 = (ND <9):
     C$(1) =  STR$ (ND):
     & B(20 * (ND = 9),20):
     PRINT B$(A1);:
     FLASH :
```

```
          PRINT C$(A1);:
          NORMAL :
          PRINT LEFT$ (" OF 8",1 + 4 * A1):
          SCALE= 1:
          XDRAW 67 AT FN PX(X(N)), FN PY(Y(N)):
          PRINT "( S D Q LEFT ARROW RIGHT ARROW )";:
          GET A$:
          XDRAW 67 AT FN PX(X(N)), FN PY(Y(N))

122   A = ASC (A$):
      N = N + FN S(NP):
      ON A$ = "Q" GOTO 124:
      ON A$ = "D" GOSUB 26:
      ON NOT (A$ = "S" AND ND <9) GOTO 120:
      GOSUB 24:
      & B(A1 * 50,50):
      ON A1 GOTO 120:
      GN%(RE,ND) = N:
      XDRAW 65 AT FN PX(X(N)), FN PY(Y(N)):
      ND = ND + 1:
      A = 21:
      N = N + FN S(NP):
      GOTO 120
```

Assure that all 8 points for a region have been defined.

```
124   N = 0:
      FOR J = 1 TO 8:
          N = N + (GN%(RE,J) >0):
      NEXT J:
      GN%(RE,0) = RE * (N = 8):
      ON N = 8 GOTO 126:
      POKE 34,20:
      HOME :
      PRINT "[G][M]REGION IS LACKING "8 - N" POINTS":
      & I("[M]PRESS <RETURN>TO REDEFINE";"";
         A$,"19"):
      GOSUB 36:
      GOSUB 38:
      GOTO 120
```

(126-128) Define NQ and ask if more regions are to be defined.

```
126   & C(B$,C$):
      NQ = NQ * (RE < = NQ) + RE * (RE >NQ):
      GOSUB 32:
      GOSUB 36:
      GOSUB 28

128   & D(G):
      POKE 34,20:
      HOME :
      & I("ALL REGIONS DEFINED? (Y/N)";"Y";A$,"20",0,1,
         A$ = "Y" OR A$ = "N"):
      ON A$ = "Y" GOTO 132
```

Edit region.

```
130   HOME :
      PRINT "EDIT REGIONS-[M] OPTIONS: ENTER, DELETE, QUIT":
      & I("( E D Q )";"E";A$,"21",BC%,0,A$ = "E" OR A$
         = "D" OR A$ = "Q"):
      ON BC% >0 GOTO 128:
      RE = RE * (RE <10) + (A$ = "E")
         + (RE <1 OR RE >9):
      ON A$ = "E" GOTO 118:
```

```
       IF A$ = "D" THEN
          GOSUB 40:
          GOTO 128
```

Determine connectivity of regions.

```
132    TEXT :
       & D(T):
       HOME :
       PRINT "CONNECTIVITY OF REGIONS[M][M]
         COMMON SIDES AND POINTS WILL BE[M]
         IDENTIFIED AUTOMATICALLY.":
       GOSUB 6:
       PRINT "[M][M]DETERMINING CONNECTIVITY...";:
       GOSUB 174:
       A1 = A1 * IG + (3 - A1) * NOT IG:
       ON A1 GOTO 128,162,164
```

(134-146) Specify the number of nodes on each side of the region.

```
134    TEXT :
       & D(T):
       HOME :
       PRINT "GEOMETRY: NODES PER SIDE  (1.33)[M][M]INSTRUCTIONS:[M]
         THE NUMBER OF TRIANGULAR  ELEMENTNODES ON
         EACH CURVILINEAR QUADRILATERALREGION SIDE
         (INCLUDING BOTH END NODES)IS >=2. OPPOSITE
         REGION SIDES MUST HAVE"

136    PRINT "THE SAME NUMBER OF NODES;
         THIS IS AUTO- MATICALLY ENFORCED. [M][M]
         USE ARROWS TO POSITION THE CURSOR.
         THEN PRESS <S>TO SELECT."

138    PRINT "[M]ENTER THE NUMBER OF NODES FOR THE MARKEDSIDES.
        CHANGE ENTRIES BY RE-SELECTING  THE SIDE.":
       GOSUB 6:
       ON BC% GOSUB 206:
       ON BC% GOSUB 218

140    ON BC% GOTO 114,148,134:
       SI%(0) = 1:
       VTAB 21:
       & D(G):
       & C(NC%,NR%,NS%):
       DIM NC%(10),NR%(10),NS%(2):
       DEF FN WN(N) = 2 * ((N <3) - (N >2)):
       POKE 34,20:
       HOME :
       PRINT "DEFINE NUMBER OF NODES PER SIDE-":
       POKE 34,21:
       N = 2:
       RE = 1:
       NN = 4

142    SCALE= 2:
       XDRAW 67 AT FN PX(X(GN%(RE,2 * N))),
         FN PY(Y(GN%(RE,2 * N))):
       HOME :
       PRINT "OPTIONS: SELECT SIDE, QUIT[M]
         ( LEFT ARROW  RIGHT ARROW S Q) ";:
       GET A$:
       A = ASC (A$):
       XDRAW 67 AT FN PX(X(GN%(RE,2 * N))),
         FN PY(Y(GN%(RE,2 * N))):
       SCALE= 1
```

```
144  N = N +  FN S(NN):
     RE = RE +  (N = 2 AND A = 21) - (N = 2 AND A = 8):
     RE = RE * (RE < = NQ AND RE >0)
        +  (RE >NQ) +  NQ * (RE <1):
     ON A$ = "Q" GOTO 146:
     ON A$ = "S" GOSUB 42:
     GOTO 142

146  NF = 0:
     FOR I = 1 TO NQ:
        NF = NF +  (NR%(I) = 0) +  (NC%(I) = 0):
     NEXT I:
     ON NF = 0 GOTO 148:
     HOME :
     FLASH :
     PRINT "[G]NUMBER OF NODES DEFINITION IS INCOMPLETE";:
     NORMAL :
     GOSUB 4:
     GOTO 142
```

Completion of data entry.

```
148  POKE 2165,2:
     POKE 2166,5:
     TEXT :
     & D(T):
     HOME :
     PRINT "[M][M]*****  END OF DATA ENTRY  *****":
     A1 = IG +  2 *  NOT IG:
     GOSUB 4:
     ON (BC% = 1) * A1 GOTO 134,170:
     GOSUB 186:
     ON (BC% = 1) * A1 GOTO 134,170:
     RETURN
```

(150-154) Enter coordinates of points (without graphics).

```
150  HOME :
     PRINT "COORDINATES OF POINTS DEFINING INPUT[M]
        QUADRILATERAL REGIONS (1.11)[M][M]
        DEFINE ENDS AND 1 INTERMEDIATE POINT FOREACH SIDE[M]
        > STOPS POINT DEF'N[M]
        < RETURNS TO PREVIOUS PT[M][M]POINT   COORDINATE":
     NO = 1:
     POKE 34,7:
     SI%(0) = 1:
     SI%(1) = 0

152  PRINT "[M]  "NO;:
     HTAB 10:
     & I(DF$(1) +  "=";X(NO);X(NO),"22",BC%):
     NO = NO - (BC% = 1 AND NO >1):
     ON BC% GOTO 152,154,152:
     HTAB 10:
     & I(DF$(2) +  "=";Y(NO);Y(NO),"23",BC%):
     A = (BC% = 0):
     ON BC% = 1 OR BC% = 3 GOTO 152:
     NP = NP +  A * (NO >NP):
     NO = NO +  A * (NO <100):
     ON A GOTO 152

154  ON ED GOTO 170:
     & I("[M]EDIT COORDINATES","N";A$,"24",0,1):
     A = (A$ = "Y"):
     ON A = 0 GOTO 112:
```

```
    ON A = 1 AND IG = 1 GOTO 108:
    & I("START AT POINT NO. ";1;NO,"25",
       BC%,0,NO >0 AND NO < = NP):
    ON BC% GOTO 154,112,154:
    HOME :
    GOTO 152
```

(156-162) Select 8 region points in a counterclockwise order (without graphics).

```
156 TEXT :
    & D(T):
    HOME :
    PRINT "POINT #S FOR EACH INPUT REGION (1.21)[M]
       8 POINTS CCW FROM SIDE 1:[M]
       (8 TIMES ENTER NUMBER <RETURN>)[M]
       <DECREMENTS PT NO.[M]  >EDIT POINTS[M][M]
       REGION      POINT NUMBERS":
    NO = 1:
    POKE 34,7

158 PRINT "[M]  "NO;:
    FOR J = 1 TO 8:
        HTAB 10:
        A = GN%(NO,J):
        & I("#" +  STR$ (J) +  "     POINT #";A;A,"26",
           BC%,0,A < = NP AND A >0):
        GN%(NO,J) = A:
        J = J - (1 +  (J >1)) * (BC% = 1) +  8
           * (BC% = 2) - (BC% = 3):
    NEXT J

160 A = (BC% = 0):
    NQ = NQ +  A * (NO >NQ):
    GN%(NO,0) = A * NO:
    NO = NO +  A:
    ON A GOTO 158:
    & I(R$ +  "REGIONS";NQ;NQ,"27")

162 ON ED GOTO 170:
    & I("[M]EDIT INPUT REGION POINT #S";"N";A$,"28",0,1):
    ON A$ < >"Y" GOTO 132:
    & I("START AT REGION NO. ";NO;NO,"29",
       BC%,0,NO >0 AND NO < = NQ):
    ON BC% GOTO 162,132,162:
    HOME :
    GOTO 158
```

(164-168) Specify the number of nodes per region side (without graphics).

```
164 TEXT :
    & D(T):
    HOME :
    PRINT "DESIRED SUB-DIVISION OF REGIONS,
       I.E.,  NO. OF NODES PER INPUT REGION SIDE[M][M]
       REGION  NO. OF NODES DESIRED PER SIDE":
    POKE 34,4:
    NO = 1

166 PRINT "[M]  "NO;:
    HTAB 9:
    & I("SIDES 2,4: ";NR%(NO);NR%(NO),"30",
       BC%,0,NR%(NO) > = 2):
    NO = NO - (BC% = 1 AND NO >0):
    ON BC% GOTO 166,168,166:
    HTAB 9:
```

```
      & I("SIDES 1,3: ";NC%(NO);NC%(NO),"31",
         BC%,0,NC%(NO) > = 2):
      ON BC% GOTO 166,168,166:
      ON NO > = NQ GOTO 168:
      NO = NO +  1:
      GOTO 166

 168  ON ED = 1 GOTO 170:
      & I("[M]EDIT NODES PER SIDE","N";A$,"32",0,1):
      IF A$ = "Y" THEN
         & I("START AT REGION NO. ";NO;NO,"33",
            BC%,0,NO >0 AND NO < = NQ):
         ON BC% GOTO 168,170,168:
         HOME :
         GOTO 166
```

(170-178) Edit options.

```
 170  ED = 0:
      TEXT :
      & D(T):
      HOME :
      PRINT "GEOMETRY: EDIT (1.41)[M][M]OPTIONS:[M][M]
         1. POINT "DF$(1)"-"DF$(2)" COORDINATES[M]
         2. REGION POINT NO.S[M]
         3. NO. OF NODES PER REGION SIDE[M][M]0. NONE[M]"

 172  & I(SR$,0;NO,"34",BC%,0,NO > = 0 AND NO <4):
      ON BC% GOTO 170,148,170:
      ON NO = 0 GOTO 148:
      SI%(0) = 1:
      ED = 1:
      ON NO GOTO 150,156,164:
      GOTO 170
```

(174-176) Determine connectivity of regions.

```
 174  A1 = 0:
      FOR RE = 1 TO NQ:
         FOR J = 1 TO 4:
            JT%(RE,J) = 0:
      NEXT J,RE:
      & B(0,1,10):
      ON NQ = 1 GOTO 178:
      FOR RE = 1 TO NQ:
         FOR I = 1 TO RE - 1:
            FOR J = 2 TO 8 STEP 2:
               FOR K = 2 TO 8 STEP 2:
                  IF GN%(I,K) = GN%(RE,J) AND I
                     < >RE THEN
                     JT%(RE,J / 2) = I:
                     JT%(I,K / 2) = RE

 176           NEXT K,J:
            PRINT ".";:
      NEXT I,RE:
      RE = NQ:
      FOR I = 1 TO NQ:
         A1 = 0:
         FOR J = 1 TO 4:
            A1 = A1 +  (JT%(I,J) = 0):
         NEXT J:
         I = I +  NQ * (A1 = 4):
      NEXT I:
      A1 = (A1 = 4):
      ON A1 = 0 GOTO 178:
```

```
        PRINT "[M][M][G][G]!! WARNING !![M][M]
           A REGION HAS NO SIDE IN COMMON WITH[M]ANOTHER REGION!":
        GOSUB 4

178     HOME :
        GOSUB 310:
        A1 = (A1 +  (BC% >0) >0):
        ON A1 = 0 GOTO 58:
        & I("[M]EDIT INPUT REGIONS";"Y";A$,"35",BC%,1):
        ON BC% >0 GOTO 174:
        A1 = (A$ = "Y"):
        RETURN
```

(180-184) Save coordinates of region points and disable subsequent files.

```
180     TEXT :
        & D(T):
        HOME :
        & C(XY):
        DIM XY(NP,2):
        FOR I = 1 TO NP:
            XY(I,0) = I:
            XY(I,1) = X(I):
            XY(I,2) = Y(I):
        NEXT I:
        RE = 1:
        GOSUB 230:
        GOSUB 224:
        SI%(1) = 0:
        N1 = NP:
        N2 = 2:
        ONERR  GOTO 252

182     EF = 0:
        GOSUB 226:
        ON  PEEK (2163) = 3 GOTO 184:
        & S(XY,N$)

184     ON EF GOTO 182:
        GOSUB 228:
        R1 = 2:
        R2 = 22:
        GOSUB 250:
        RETURN
```

(186-190) Save region definitions and disable subsequent files.

```
186     & C(RD%):
        DIM RD%(NQ,14):
        FOR I = 1 TO NQ:
            RD%(I,0) = I:
            FOR J = 1 TO 8:
                RD%(I,6 +  J) = GN%(I,J):
            NEXT J:
            FOR J = 1 TO 4:
                RD%(I,J) = JT%(I,J):
            NEXT J:
            RD%(I,5) = NR%(I):
            RD%(I,6) = NC%(I):
        NEXT I:
        RE = 2:
        GOSUB 224:
        N1 = NQ:
        N2 = 14:
        ONERR  GOTO 252
```

```
188  EF = 0:
     TEXT :
     & D(T):
     HOME :
     GOSUB 226:
     ON  PEEK (2163) = 3 GOTO 190:
     & S(RD%,N$)

190  ON EF GOTO 188:
     GOSUB 228:
     R1 = 3:
     R2 = 22:
     GOSUB 250:
     GOTO 4
```

(192-194) Retrieve region coordinate points and find extremes.

```
192  RE = 1:
     GOSUB 230:
     GOSUB 224:
     ON SI%(1) = 0 GOTO 200:
     ONERR  GOTO 252

194  EF = 0:
     GOSUB 234:
     NP = N1:
     & C(XY):
     DIM XY(NP,2):
     & R(XY,N$):
     ON EF GOTO 194:
     GOSUB 242:
     FOR I = 1 TO NP:
          X(I) = XY(I,1):
          Y(I) = XY(I,2):
     NEXT I:
     & O(XY,1 TO NP,1,A):
     XL = XY(1,1):
     XH = XY(NP,1):
     & O(XY,1 TO NP,2,A):
     YL = XY(1,2):
     YH = XY(NP,2):
     & C(XY):
     RETURN
```

(196-198) Retrieve region definitions.

```
196  RE = 2:
     GOSUB 230:
     GOSUB 224:
     ON SI%(RE) = 0 GOTO 200:
     & C(RD%)

198  EF = 0:
     GOSUB 234:
     NQ = N1:
     DIM RD%(NQ,14):
     & R(RD%,N$):
     ON EF GOTO 198:
     GOSUB 242:
     FOR I = 1 TO NQ:
          GN%(I,0) = I:
          FOR J = 1 TO 8:
               GN%(I,J) = RD%(I,6 +  J):
          NEXT J:
          FOR J = 1 TO 4:
               JT%(I,J) = RD%(I,J):
```

```
        NEXT J:
        NR%(I) = RD%(I,5):
        NC%(I) = RD%(I,6):
    NEXT I:
    RETURN
```

Warn of invalid data.

```
200 FLASH :
    PRINT "ERROR # "; PEEK (222):
    PRINT "[G][G][M]DATA NOT DEFINED":
    NORMAL :
    PRINT "EITHER FILE":
    INVERSE :
    PRINT NA$:
    NORMAL :
    PRINT "DOESN'T EXIST OR THE DATA IS NOT VALID":
    GOTO 4
```

(202-204) Draw label L$ on screen.

```
202 ON  LEN (L$) >5 GOTO 204:
    FOR I1 = 1 TO  LEN (L$):
       DRAW  ASC ( MID$ (L$,I1,1)) - 31 AT  FN PX(XP)
         - 3.5 *  LEN (L$) +  7 * (I1 - 1) +  XA * (3.5
         *  LEN (L$) +  5), FN PY(YP) +  3.5 - YA * 5:
    NEXT I1:
    RETURN

204 LL =  LEN (L$) - 5:
    L1$ =  RIGHT$ (L$,LL):
    L$ =  LEFT$ (L$,5):
    GOSUB 202:
    YP =  FN VY( FN PY(YP) +  8):
    L$ = L1$:
    GOTO 202
```

(206-216) Draw and label axes.

```
206 HGR :
    & D(G):
    HCOLOR= 3:
    SCALE= 1:
    X1 =  FN PX(UL):
    Y1 =  FN PY(VL) +  5:
    X2 =  FN PX(UH):
    DV = (X2 - X1) / 10:
    ROT= 16:
    DRAW 75 AT X1,Y1:
    IF X2 - SL >40 THEN
         FOR I = 1 TO 9:
            X1 = X1 +  DV:
            HPLOT X1,Y1:
    NEXT I

208 DRAW 75 AT X2,Y1:
    DRAW 75 AT SR,SB +  5:
    D = 3:
    ROT= 0:
    L$ =  STR$ ( FN RS(UL)):
    XA = 0:
    YA =  - 2.4:
    XP = UL:
    YP = VL:
    GOSUB 202
```

```
210  XP =  FN UX(SR):
     L$ =  STR$ ( FN RS(XP)):
     XA =  - .7:
     YP = VL:
     GOSUB 202:
     L$ =  STR$ ( FN RS(UH)):
     XA = 0:
     XP = UH:
     ON (X2 - 3.5 *  LEN (L$) >SL +  15) AND (X2
        +  3.5 *  LEN (L$) <SR - 28) GOSUB 202:
     X1 =  - 30 * (X2 >130 AND X2 <SR - 28)
        +  30 * (X2 <131 AND X2 >SL +  15):
     DRAW 70 AT 130 +  X1,155

212  DRAW  ASC (DF$(1)) - 31 AT 148 +  X1,158:
     X1 =  FN PX(UL) - 5:
     Y1 =  FN PY(VL):
     Y2 =  FN PY(VH):
     DV = (Y2 - Y1) / 10:
     ROT= 32:
     DRAW 75 AT X1,Y1:
     IF SB - Y2 >40 THEN
         FOR I = 1 TO 9:
            Y1 = Y1 +  DV:
            HPLOT X1,Y1:
     NEXT I

214  DRAW 75 AT X1,Y2:
     DRAW 75 AT SL - 5,ST:
     ROT= 0:
     L$ =  STR$ ( FN RS(VL)):
     XP = UL:
     YP = VL:
     XA =  - 1.4:
     YA = 0:
     GOSUB 202:
     YP =  FN VY(ST):
     L$ =  STR$ ( FN RS(YP)):
     GOSUB 202:
     L$ =  STR$ ( FN RS(VH)):
     YP = VH:
     ON Y2 >ST +  8 AND Y2 <SB - 10 GOSUB 202

216  Y1 =  - 15 * (Y2 >90 AND Y2 <SB - 10)
        +  15 * (Y2 <91 AND Y2 >ST +  8):
     DRAW  ASC (DF$(2)) - 31 AT 2,75 +  Y1:
     ROT= 48:
     DRAW 70 AT 5,90 +  Y1:
     ROT= 0:
     RETURN
```

Draw a '+' symbol at all input points.

```
218  ON NP = 0 GOTO 58:
     SCALE= 2:
     FOR I = 1 TO NP:
        XDRAW 64 AT  FN PX(X(I)), FN PY(Y(I)):
     NEXT I:
     SCALE= 1:
     RETURN
```

Draw boundary and label all regions.

```
220  & D(G):
     DF = 1:
```

```
     FOR RE = 1 TO NQ:
          GOSUB 32:
          GOSUB 28:
          ON Z = 3 GOSUB 4:
     NEXT RE:
     RE = NQ:
     VTAB 21:
     POKE 34,20:
     HOME :
     RETURN
```

Define scale and allowable error for plotting.

```
222  AE = 1E - 10 * ((UH - UL) ^ 2 + (VH - VL) ^ 2):
     S0 = (SB - ST) / (VH - VL):
     SC = (SR - SL) / (UH - UL):
     SC = SC * (SC < = S0) + S0 * (SC >S0):
     S0 = SC:
     RETURN
```

Read information about file RE.

```
224  N1 = FRE (0):
     PRINT D$:
     PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"READ"FI$",R"RE:
     INPUT SI%(RE),NA$,DE$,N1,D1$,N2,D2$:
     PRINT D$"CLOSE"FI$:
     N$ = NA$ + DR$(DR):
     RETURN
```

Request filename for data to be saved.

```
226  ON Z = 2 OR PEEK (2163) = 3 GOTO 236:
     PRINT "[M]*** ";:
     INVERSE :
     PRINT "SAVING";:
     NORMAL :
     PRINT " DATA TO DISKETTE ***[M][M]":
     ON Z GOTO 240:
     PRINT "FILE NAME FOR "DE$:
     & !("";NA$,NA$,"36",BC%,30):
     ON BC% >0 GOTO 226:
     GOSUB 238::
     N$ = NA$ + DR$(DR):
     RETURN
```

Save information about file being saved.

```
228  SI%(RE) = 1:
     ON PEEK (2163) = 3 GOTO 58:
     PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"WRITE"FI$",R"RE:
     PRINT 1:
     PRINT NA$:
     PRINT DE$:
     PRINT N1:
     PRINT D1$:
     PRINT N2:
     PRINT D2$:
     PRINT D$"CLOSE"FI$:
     PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2" "D2$"[M]
        WERE SAVED":
     RETURN
```

(230-232) Verify the existence of data file.

```
230  ONERR GOTO 262
```

```
232  E2 = 0:
     PRINT D$:
     PRINT D$"CKFILE"FI$;DR$(DR):
     ON E2 GOTO 232:
     RETURN
```

(234-242) Print information about data file being loaded.

```
234  ON Z = 2 GOTO 236:
     PRINT "[M]*** ";:
     INVERSE :
     PRINT "LOADING";:
     NORMAL :
     PRINT " DATA FROM DISKETTE ***[M][M]":
     ON Z GOTO 240:
     PRINT "FILE NAME FOR":
     & I(DE$,NA$,NA$,"37",BC%,30):
     ON BC% >0 GOTO 234:
     GOSUB 238

236  N$ = NA$ +  DR$(DR):
     RETURN


238  ON  LEFT$ (NA$, LEN (KW$)) = KW$ GOTO 58:
     NA$ = KW$ +  "/" +  NA$:
     RETURN

240  PRINT " "NA$:
     RETURN

242  ON Z = 2 GOTO 58:
     PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2""
        D2$"[M]WERE LOADED":
     RETURN
```

(244-246) Read and print problem description.

```
244  PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"READ"FI$",R0":
     INPUT TE,KW$,PD$:
     PRINT D$"CLOSE"FI$:
     RETURN

246  ON  LEN (KW$) = 0 GOSUB 244:
     PRINT "[M]PROBLEM KEYWORD: "KW$"[M][M]
        PROBLEM DESCRIPTION:[M]"PD$:
     RETURN
```

Update problem description in record 0.

```
248  ON  PEEK (2163) = 3 GOTO 58:
     PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"WRITE"FI$",R0":
     PRINT TE:
     PRINT KW$:
     PRINT PD$:
     PRINT D$"CLOSE"FI$:
     PRINT "[M][M][M][M] ** ELEMENT TYPE "TE" SAVED **":
     GOTO 4
```

Disable data files.

```
250  ON  PEEK (2163) = 3 GOTO 58:
     PRINT D$"OPEN"FI$",L100";DR$(DR):
     FOR RE = R1 TO R2:
        SI%(RE) = 0:
```

```
      PRINT D$"WRITE"FI$",R"RE:
      PRINT 0:
   NEXT RE:
   PRINT D$"CLOSE"FI$:
   RETURN
```

(252-262) Error traps.

```
252 EF = 1:
    ER = PEEK (222):
    POKE 216,0:
    ON ER = 255 GOTO 256:
    FLASH :
    PRINT "[G][G][M]ERROR # ";ER;"[M]":
    NORMAL :
    ER = ER * (ER >3 AND ER <11)
       +   NOT (ER >3 AND ER <11):
    ON ER = 1 OR ER = 5 GOTO 254:
    ON ER >3 AND ER <11 GOTO 258

254 GOSUB 224:
    RESUME

256 END

258 PRINT ER$(ER - 4):
    ON ER = 6 GOSUB 260:
    ON ER = 4 OR ER = 5 OR ER = 8 OR ER = 9 GOSUB 4:
    ON ER <10 GOTO 254:
    & I("UNLOCK? (Y/N)";"Y";A$,"38"):
    ON  LEFT$ (A$,1) = "N" GOTO 254:
    PRINT "UNLOCKING FILE":
    PRINT D$"UNLOCK "NA$DR$(DR):
    GOTO 254


260 & I("[M]CATALOG? (Y/N)";"N";A$,"39"):
    ON  LEFT$ (A$,1) < >"Y" GOTO 58:
    PRINT D$"CATALOG"DR$(DR):
    RETURN

262 E2 = 1:
    POKE 216,0:
    ONERR  GOTO 252:
    PRINT "[G][G][M]DATA DISKETTE IS NOT IN DRIVE":
    GOSUB 4:
    RESUME
```

(264-280) Main program entry and initialization.

```
264 HOME :
    INVERSE :
    FOR I = 1 TO 4:
       PRINT  TAB( 2)" " TAB( 38)" ":
    NEXT I:
    NORMAL :
    VTAB 2:
    HTAB 3:
    PRINT " APPLIED FINITE ELEMENT ANALYSIS: ":
    HTAB 3:
    PRINT "     AN APPLE " CHR$ (221) CHR$ (219)
       " IMPLEMENTATION      ":
    VTAB 6:
    HTAB 12:
    PRINT "COPYRIGHT, 1985[M]"
```

```
266  HTAB 12:
     INVERSE :
     PRINT "GEOMETRY";:
     NORMAL :
     PRINT "  (1.00)[M]":
     & C(ER$):
     DIM ER$(6):
     ER$(0) = "DISKETTE IS WRITE PROTECTED":
     ER$(2) = "FILE NOT FOUND":
     ER$(3) = "VOLUME MISMATCH":
     ER$(4) = "I/O ERROR - DOOR/DISK INIT":
     ER$(5) = "DISK IS FULL"

268  ER$(6) = "FILE IS LOCKED":
     PRINT "ABSTRACT:[M][M]DISCRETIZATION BEGINS WITH
        THE CREATION OF CURVILINEAR QUADRILATERAL
        SUB-REGIONSWHICH TOGETHER DESCRIBE THE OVERALL[M]
        GEOMETRY.[M][M]SUBSEQUENTLY, THESE INPUT REGIONS
        WILL  BE SUB-DIVIDED INTO A MESH";

270  PRINT " OF 3-NODE[M]TRIANGLES.":
     GOSUB 6:
     & C(DR$,UP$,I$):
     DIM DR$(2),UP$(1),I$(1):
     UP$(1) = " ( : FOR OLD PT)":
     DR$(1) = ",D1":
     DR$(2) = ",D2":
     I$(0) = " INPUT WITH GRAPHICS ":
     I$(1) = " INPUT WITHOUT GRAPHICS "

272  DR =  PEEK (2048):
     IF DR = 1 THEN
          & B(200,20):
         PRINT "INSERT ";:
         FLASH :
         PRINT "DATA";:
         NORMAL :
         PRINT " DISKETTE INTO DRIVE":
         GOSUB 4

274  GOSUB 244:
     HOME :
     PRINT "ELEMENT TYPE?[M][M]
        1. TWO-DIMENSIONAL 3-NODE TRIANGLE[M]
        2. AXISYMMETRIC 3-NODE TRIANGLE[M]":
     & I(SR$,TE;TN,"40",BC%,0,TN = 1 OR TN = 2):
     ON BC% GOTO 264,274,274:
     ON Z = 3 AND TN < >TE GOSUB 320:
     IF TN < >TE THEN
        TE = TN:
        GOSUB 230:
        GOSUB 248:
        ON BC% >0 GOTO 274

276  ST = 10:
     SB = 140:
     SL = 50:
     SR = 275:
     DEF  FN PX(X) = SL +  SC * (X - UL):
     DEF  FN PY(Y) = SB - SC * (Y - VL):
     DEF  FN UX(X) = UL +  (X - SL) / SC:
     DEF  FN VY(Y) = VL - (Y - SB) / SC:
     & C(X,Y,JT%,NR%,NC%,GN%,PN$):
     DIM X(100),Y(100)
```

```
278  & C(SI%,DI$,DF$,AC$,LD$):
     DIM SI%(24),DI$(4),DF$(2),AC$(2),LD$(3),JT%(10,4),
        NR%(10),NC%(10),GN%(11,8),PN$(1):
     LG =  LOG (10):
     DEF  FN RS(X) =  SGN (X) *  INT ( ABS (X)
        * 10 ^ (D - 1 -  INT ( LOG ( ABS (X +  (X
        = 0))) / LG)) +  .5) / (10 ^ (D - 1
        -  INT ( LOG ( ABS (X +  (X = 0))) / LG)))

280  DI$(1) = "X":
     DI$(2) = "Y":
     DI$(3) = "R":
     DI$(4) = "Z":
     DF$(1) = DI$(2 * TE - 1):
     DF$(2) = DI$(2 * TE):
     AC$(1) = "DELETE":
     AC$(2) = "ENTER":
     LD$(1) = "SINGLE POINT":
     LD$(2) = "STRAIGHT LINE":
     LD$(3) = "CIRCULAR ARC":
     PN$(0) = "HELLO":
     PN$(1) = "GRID"
```

(282-284) Restart entry and menu.

```
282  & L(255):
     NM =  PEEK (2166):
     TEXT :
     & D(T):
     HOME :
     EO = 1:
     ON Z = 2 AND  PEEK (2165) = 2 GOTO 288:
     PRINT "***         GEOMETRY (1.05)         ***[M][M]
        --- POINT COORDINATE ENTRY ---[M]
        1."I$(0)"[M]
        2."I$(1)"[M][M]
        --- REGION DEFINITION/DIVISION  ---[M]
        3."I$(0)"[M]
        4."I$(1)

284  PRINT "[M]
        ---  DATA REVIEW   ---[M]
        5. LIST INPUT DATA[M]
        6. PLOT INPUT DATA[M][M]
        0. NONE OF THE ABOVE[M]":
     & I(SR$,NM;MO,"41",BC%,0,MO > = 0 AND MO <7):
     ON BC% GOTO 274,286,282:
     ON MO = 0 GOTO 286:
     IG = (MO = 1 OR MO = 3):
     ON MO GOSUB 292,294,292,294,302,316:
     GOTO 282
```

(286-290)Exit menu.

```
286  HOME :
     PRINT "***        GEOMETRY: EXIT  (1.70)        ***[M]
        1. PROCEED TO 'GRID'[M]
        2. EXIT TO MAIN MENU[M]
        3. DON'T EXIT; REMAIN IN 'GEOMETRY'[M][M]
        0. STOP[M][M]":
     EO = 1:
     & I(SR$,1;EO,"42",BC%,1,EO > = 0 AND EO <4):
     ON EO = 0 OR BC% = 2 GOTO 256:
     ON EO = 3 GOTO 282
```

```
288 POKE 2166,1:
    IF DR = 1 THEN
         & B(200,20):
        PRINT "[M]INSERT ";:
        FLASH :
        PRINT "PROGRAM";:
        NORMAL :
        PRINT " DISKETTE CONTAINING[M]'";:
        INVERSE :
        PRINT PN$(EO = 1);:
        NORMAL :
        PRINT "' INTO DRIVE ":
        GOSUB 4:
        ON BC% = 1 OR BC% = 3 GOTO 286

290 PRINT D$"RUN "PN$(EO = 1)",D1"
```

(290-300) Menu option 1 and 3: Input with graphics: points, regions and nodes/side.

```
292 DEF FN S(NP) = (A = 21) * ((N < >NP)
      - (NP - 1) * (N = NP)) - (A = 8)
      * ((N < >1) - (NP - 1) * (N = 1)):
      DEF FN A(N) = ATN ((Y(N) - YC) / (X(N) - XC
      + 1E10 * (X(N) = XC))) + 180 * RD
         * (X(N) <XC)
      + SGN (Y(N) - YC) * 90 * RD * (X(N) = XC):
    RD = ATN (1) / 45:
    DEF FN EQ(N) = ((X(N) - X(I)) ^ 2
      + (Y(N) - Y(I)) ^ 2 <AE):
```

Menu options 2 and 4: Input without graphics.

```
294 NQ = 0:
    HOME :
    PRINT "[M]*** "I$(IG = 0)" ***[M]":
    IF MO = 1 OR MO = 2 THEN
         & C(X,Y):
        DIM X(100),Y(100):
        NP = 0:
        SI%(1) = 0:
        & I("RETRIEVE COORDINATE DATA FROM DISK";"N";
           A$,"43",BC%,1):
        ON BC% >0 GOTO 58:
        ED = (A$ = "Y"):
        ON ED GOSUB 192:
        ON (MO = 2) * (ED + 1) GOTO 150,154:
        ED = 0

296 IF MO = 3 OR MO = 4 THEN
         ON SI%(1) = 0 GOSUB 192:
        UL = XL:
        UH = XH:
        VL = YL:
        VH = YH:
        GOSUB 222:
        & I("[M]RETRIEVE REGION DATA FROM DISK";"N";
           A$,"44",BC%):
        ON BC% >0 GOTO 58:
        ED = ( LEFT$ (A$,1) = "Y"):
        ON ED GOSUB 196:
        ON BC% >0 GOTO 58:
        ON MO = 3 GOTO 114:
        ON ED + 1 GOTO 156,170

298 & I("[M][M] -- SET AXIS LIMITS --[M][M]MINIMUM "
      + DF$(1) + " = ";XL;UL,"45",BC%):
```

```
     ON BC% GOTO 294,58,298:
     & I(" MAXIMUM = ";XH;UH,"46",BC%,0,UH >UL):
     ON BC% GOTO 298,58,298:
     UE = .01 * (UH - UL):
     Z1 = UL - UE:
     Z2 = UH +  UE:
     XL = UL:
     XH = UH

300  & I("[M]MINIMUM " +  DF$(2) +  " = ";YL;VL,"47",BC%):
     ON BC% GOTO 298,58,300:
     & I(" MAXIMUM = ";YH;VH,"48",BC%,0,VH >VL):
     ON BC% GOTO 300,58,300:
     VE = .01 * (VH - VL):
     Z3 = VL - VE:
     Z4 = VH +  VE:
     YL = VL:
     YH = VH:
     GOSUB 222:
     GOTO 104
```

(302-314) Menu option 5: List coordinates, regions, connectivity, and nodes/side.

```
302  HOME :
     PRINT "SUMMARY OF INPUT DATA (1.5)[M][M]":
     POKE 34,2:
     GOSUB 246:
     ON SI%(1) = 0 GOSUB 192:
     ON SI%(1) = 0 OR BC% = 2 GOTO 58:
     GOSUB 4:
     GOSUB 306:
     ON BC% >0 GOTO 58

304  ON SI%(2) = 0 GOSUB 196:
     ON SI%(2) = 0 OR BC% = 2 GOTO 58:
     GOSUB 308:
     ON BC% >0 GOTO 58:
     GOSUB 310:
     ON BC% >0 GOTO 58:
     GOSUB 314:
     RETURN

306  HOME :
     & L(P%):
     PRINT "COORDINATES OF POINTS[M]":
     INVERSE :
     PRINT "POINT      "DF$(1) TAB( 25)DF$(2) TAB( 39)" ":
     NORMAL :
     POKE 34,5:
     FOR I = 1 TO NP:
          PRINT "  "I TAB( 10)X(I) TAB( 25)Y(I):
     NEXT I:
     POKE 34,2:
     GOTO 4

308  POKE 34,2:
     HOME :
     & L(P%):
     PRINT "POINT NUMBERS BY REGION[M]":
     INVERSE :
     PRINT "REGION      GLOBAL POINT NUMBERS" TAB( 39)" ":
     NORMAL :
     POKE 34,5:
```

```
        FOR I = 1 TO NQ:
            PRINT "  "I TAB( 8);:
            FOR J = 1 TO 8:
                PRINT  LEFT$ ( STR$ (GN%(I,J)) +  "    ",4);:
            NEXT J:
            PRINT :
        NEXT I:
        POKE 34,2:
        GOTO 4

310     POKE 34,2:
        HOME :
        DIM B$(1):
        B$(0) = "A BOUNDARY":
        B$(1) = "BY REGION ":
        & L(P%):
        PRINT "REGION CONNECTIVITY DATA[M]":
        INVERSE :
        PRINT "REGION      CONNECTIVITY" TAB( 39)" ":
        NORMAL :
        POKE 34,5:
        FOR I = 1 TO NQ:
            PRINT "  "I;:
            FOR J = 1 TO 4:
                A = (JT%(I,J) < >0)

312             A$ = B$(A) +   LEFT$ (" " +   STR$ (JT%(I,J)),
                  1 +  A *  LEN ( STR$ (JT%(I,J)))):
                PRINT  TAB( 10)"SIDE "J" IS "A$:
            NEXT J:
            PRINT :
        NEXT I:
        & C(B$):
        POKE 34,2:
        GOTO 4

314     POKE 34,2:
        HOME :
        & L(P%):
        PRINT "NUMBER OF NODES PER REGION SIDE[M]":
        INVERSE :
        PRINT "REGION    SIDES 1&3        SIDES 2&4"
          TAB( 39)" ":
        NORMAL :
        POKE 34,5:
        FOR I = 1 TO NQ:
            PRINT "  "I TAB( 15)NR%(I) TAB( 30)NC%(I):
        NEXT I:
        POKE 34,2:
        POKE 2166,6:
        GOTO 4
```

(316-318) Menu option 6: Plot defined regions

```
316     HOME :
        PRINT "PLOT OF INPUT DATA (1.6)[M][M]":
        POKE 34,2:
        GOSUB 246:
        GOSUB 192:
        ON SI%(1) = 0 GOTO 58:
        GOSUB 196:
        ON SI%(2) = 0 OR BC% >0 GOTO 58:
        GOSUB 4
```

```
318  UL = XL:
     UH = XH:
     VL = YL:
     VH = YH:
     GOSUB 222:
     ON BC% = 1 GOTO 58:
     GOSUB 206:
     GOSUB 220:
     PRINT "[M]REGIONS DEFINED BY INPUT DATA":
     GOSUB 4:
     TEXT :
     & D(T):
     RETURN
```

Warn that demo mode cannot change type.

```
320  PRINT "[G][G]CORRECTION:[M]DEMO DATA APPLIES TO TYPE "TE:
     GOSUB 4:
     TN = TE:
     RETURN

65535 REM [M][M]1FEB85[M]JRC/DCD

END-OF-LISTING
```

GRID.HEAT

Cold start (initialize).

```
1  & L(255):
   D$ =  CHR$ (4):
   SR$ = "... SELECT BY NUMBER <RETURN>":
   Q$ = "( LEFT ARROW  RIGHT ARROW  Q )":
   S$ = "( LEFT ARROW  RIGHT ARROW  S Q )":
   M$ = "MOVE CURSOR USING LEFT AND RIGHT ARROWS.
     TYPE <Q>TO QUIT ":
   POKE 2164,2:
   Z =  PEEK (2163):
   GOTO 240:
   REM GRID(H/E)
```

Warm start (restart entry).

```
2  & L(255):
   GOTO 256
```

Wait for user response, clear screen and continue.

```
4  ON FR GOSUB 268:
   PRINT :
   VTAB 24 * (FR = 0) +  ( PEEK (37) +  1) * (FR = 1):
   CALL  - 958:
   ON FR GOTO 20:
   & B(20,20):
   & I("... WHEN READY, PRESS <RETURN>";
     "";A$,"1",BC%):
   HOME :
   RETURN
```

Remove grid lines to selected mode, move node and redraw mesh lines.

```
6  J1 = ND%(N,1):
   J2 = ND%(N,2):
   J3 = ND%(N,3):
   HPLOT  FN PX(XN(J1)), FN PY(YN(J1)) TO  FN PX(XN(J2)),
     FN PY(YN(J2)) TO  FN PX(XN(J3)),
     FN PY(YN(J3)) TO  FN PX(XN(J1)),
     FN PY(YN(J1)):
   RETURN
```

(8-10) Identify nodes.

```
8  POKE 34,20:
   HOME :
   PRINT "NODE #"N"[M]"DF$(1)"="XN(N) TAB( 21)DF$(2)
     "="YN(N):
   XP =  FN PX(XN(N)):
   YP =  FN PY(YN(N)):
   XDRAW 66 AT XP,YP:
   PRINT "TYPE CTRL-T FOR INSTRUCTIONS[M]"S$;:
   GET A$:
   XDRAW 66 AT XP,YP:
   A =  ASC (A$):
   ON A = 20 GOTO 20:
   N = N +  FN S(NN)

10 ON A$ = "S" GOSUB 12:
   ON A$ < >"Q" GOTO 8:
   A = 0:
   RETURN
```

(12-16) Relocate a node.

```
12  HOME :
    PRINT "CHANGE POSITION OF NODE #"N"[M]PLEASE WAIT...":
    POKE 34,21:
    GOSUB 24:
    HCOLOR= 4:
    FOR I = 1 TO NR:
        X1 =  FN PX(XN(R%(I))):
        Y1 =  FN PY(YN(R%(I))):
        HPLOT XP,YP TO X1,Y1 TO XP,YP:
    NEXT I:
    HCOLOR= 3:
    SCALE= 3

14  HOME :
    PRINT DF$(1)"="XN(N) TAB( 21)DF$(2)"="YN(N):
    XDRAW 64 AT XP,YP:
    PRINT "( I J K M V Q )";:
    GET A1$:
    A =  ASC (A1$):
    ON A1$ = "V" GOSUB 18:
    XDRAW 64 AT XP,YP:
    XP = XP +  (A1$ = "K") - (A1$ = "J"):
    YP = YP +  (A1$ = "M") - (A1$ = "I")

16  XN(N) = UL +  (XP - SL) / SC:
    YN(N) = VL - (YP - SB) / SC:
    ON A1$ < >"Q" GOTO 14:
    SCALE= 1:
    FOR I = 1 TO NR:
        HPLOT XP,YP TO  FN PX(XN(R%(I))),
           FN PY(YN(R%(I))):
    NEXT I:
    A = 0:
    & C(R%):
    RETURN
```

New node coordinates.

```
18  HOME :
    & I(DF$(1) +  " = ";XN(N);XN(N),"2",BC%,0,
       XN(N) > = UL AND XN(N) < = UH):
    ON BC% >0 GOTO 20:
    & I(DF$(2) +  " = ";YN(N);YN(N),"3",
       BC%,0,YN(N) > = VL AND YN(N) < = VH):
    ON BC% GOTO 18,20,20:
    XDRAW 64 AT XP,YP:
    XP =  FN PX(XN(N)):
    YP =  FN PY(YN(N)):
    XDRAW 64 AT XP,YP

20  RETURN
```

(22-38) Node renumbering subroutines: generate a table of nodes connected to this one by mesh lines.

```
22  FOR N = 1 TO NN:
        A$ = ".[M]... FOR NODE #" +  STR$ (N):
        PRINT  LEFT$ (A$,1 +  18 * (PO >2));:
        GOSUB 24:
        FOR I = 1 TO NR:
            RN%(N,I) = R%(I):
        NEXT I:
        N%(N) = NR:
    NEXT N:
```

```
    & C(R%):
    RETURN
```

(24-26) Find all nodes connected to node N.

```
24  NR = 0:
    & C(R%):
    DIM R%(20):
    FOR J = 1 TO 3:
        J1 = FN P1(J):
        J2 = FN P2(J):
        & W((ND%,1 TO NE,J),N,N,F%,BR%,ER%):
        ON F% >0 GOSUB 28:
    NEXT J:
    & W((R%,1 TO NR,0),1,NN,F%,BR%,ER%):
    FOR I = BR% TO ER% - 1:
        N1 = R%(I):
        FOR J = I + 1 TO ER%:
            A = (N1 < >R%(J)):
            R%(J) = A * R%(J):
            I = J - 1:
            J = J + A * 20:
    NEXT J,I

26  & W((R%,1 TO NR,0),1,NN,F%,BR%,ER%):
    NR = F%:
    FOR I = 1 TO NR:
        R%(I) = R%(I + BR% - 1):
    NEXT I:
    & O(ND%,1 TO NE,4,A):
    RETURN
```

Add node numbers to array of related nodes.

```
28  FOR K = 1 TO F%:
        R%(NR + 1) = ND%(BR% + K - 1,J1):
        R%(NR + 2) = ND%(BR% + K - 1,J2):
        NR = NR + 2:
    NEXT K:
    RETURN
```

(30-38) Renumber nodes using the table of related nodes and find new bandwidth.

```
30  FOR I = I1 TO I2:
        FOR K = 1 TO NN:
            NW%(K) = 0:
            OL%(K) = 0:
        NEXT K:
        W% = 0:
        NW%(I) = 1:
        OL%(NW%(I)) = I:
        NN% = 1:
        IF PO = 1 THEN
            POKE 34,4:
            HOME :
            POKE 34,2:
            VTAB 5:
            PRINT "CALCULATING ("I" OF "NN")";

32      IF PO >1 THEN
            & L(P%):
            HOME :
            VTAB 5:
            PRINT "START NUMBERING AT OLD NODE #";:
            INVERSE :
            PRINT I:
            NORMAL :
```

```
          PRINT ".";:
          IF PO >2 THEN
               PRINT "CHANGE OLD #"I TAB( 16)"  TO #1"

34    FOR J = 1 TO NN:
          FOR K = 1 TO N%(OL%(J)):
               N1 = RN%(OL%(J),K):
               IF NW%(N1) = 0 THEN
                    NN% = NN% + 1:
                    NW%(N1) = NN%:
                    OL%(NN%) = N1:
                    PRINT ".";:
                    IF PO >2 THEN
                         PRINT "CHANGE OLD #"
                              OL%(NN%) TAB( 16)"  TO #"NN%

36             B% = ABS (J - NN%) + 1:
               A1 = (B% >W%):
               W% = W% * NOT A1 + B% * A1:
               A = (NN = NN% OR B% >BW%):
               K = K + NN * A:
          NEXT K:
          J = J + NN * A:
      NEXT J:
      IF W% <BW% THEN
          BI = I:
          BW% = W%:
          IF PO >0 THEN
               POKE 34,3:
               PRINT :
               VTAB 4:
               PRINT "     --- IMPROVED BANDWIDTH = "
                    BW%" ---":
               POKE 34,5:
               & B(10,10):
               ON Z = 0 GOSUB 4:
               PRINT

38    PRINT :
      I = I + (BC% >0) * NN:
   NEXT I:
   & L(255):
   ON PO >0 GOSUB 4:
   RETURN
```

(40-48) Sort mesh lines to identify unique lines and boundary lines; order boundary lines counterclockwise.

```
40 HOME :
   & C(BL%,BN%):
   PRINT "FINDING UNIQUE GRID LINES...[M]":
   FOR J = 1 TO 3:
      AN%(J,0) = J:
      AN%(J,1) = ND%(1,J):
      AN%(J,2) = ND%(1, FN P2(J)):
      AN%(J,3) = 1:
   NEXT J:
   NL = 3:
   FOR EL = 2 TO NE:
      FOR LI = 1 TO 3:
          N1 = ND%(EL,LI):
          N2 = ND%(EL, FN P2(LI))

42        & W((AN%,1 TO NL,1),N2,N2,F%,BR%,ER%):
          A = (F% = 0 OR (F% = 1
```

```
                    AND AN%(BR%,2) < >N1)):
             IF F% >1 THEN
                  & W((AN%,BR% TO ER%,2),N1,N1,F%,BR%,ER%):
                A = (F% = 0)

44           ON A GOSUB 54:
             AN%(BR%,3) = AN%(BR%,3) * A:
     NEXT LI,EL:
     & W((AN%,1 TO MA,3),1,NE,F%,BR%,ER%):
     NB = F%:
     & O(AN%,BR% TO ER%,1,A):
     DIM BL%(NB,3),BN%(NB,2):
     FOR I = 0 TO NB - 1:
        FOR J = 0 TO 3:
            BL%(I + 1,J) = AN%(BR% + I,J):
     NEXT J,I:
     NS = 1:
     N1 = BL%(1,1):
     PRINT "[M]SORTING BOUNDARY NODES..."

46   N = 0:
     N2 = N1:
     BN%(1,0) = 1:
     FOR I = 1 TO NB:
        & W((BL%,1 TO NB,0),1,NL,F%,BR%,ER%):
        ON F% >1 GOSUB 50:
        VTAB 9:
        PRINT N" NODES ON "NS" BOUNDARY":
     NEXT I:
     BN%(NB,1) = BL%(BR%,1):
     BN%(NB,2) = BL%(NB,3):
     BN%(0,0) = NS:
     BN%(2 * NS,0) = NB

48   & O(AN%,1 TO MA,1,D):
     & O(AN%,1 TO NL,0,A):
     HE$ = "":
     ON PO = 2 GOSUB 4:
     ON PO = 2 GOSUB 164:
     RETURN
```

(50-52) Find next boundary line.

```
50   & W((BL%,BR% TO ER%,1),N2,N2,F%,BR%,ER%):
     ON F% = 0 GOTO 20:
     N = N + 1:
     N2 = BL%(BR%,2):
     BN%(N,1) = BL%(BR%,1):
     BN%(N,2) = BL%(BR%,3):
     BL%(BR%,0) = 0:
     BN%(2 * NS,0) = N:
     ON N1 < >N2 OR I = NB GOTO 20:
     NS = NS + 1:
     BN%(2 * NS - 1,0) = N + 1

52   & W((BL%,1 TO NB,0),1,NL,F%,BR%,ER%):
     N1 = BL%(BR%,1):
     N2 = N1:
     RETURN
```

Add unique line to array.

```
54   NL = NL + 1:
     AN%(NL,0) = NL:
```

```
    AN%(NL,1) = N1:
    AN%(NL,2) = N2:
    AN%(NL,3) = EL:
    VTAB 5:
    PRINT NL" LINES ARE UNIQUE AFTER ELEMENT "EL:
    RETURN
```

Place symbol at node, display node number and coordinates and move to another node.

```
56  VTAB 21:
    HTAB 7:
    PRINT N"   ":
    HTAB 3:
    CALL  - 868:
    PRINT XN(N):
    HTAB 3:
    CALL  - 868:
    PRINT YN(N):
    XP =  FN PX(XN(N)):
    YP =  FN PY(YN(N)):
    XDRAW 66 AT XP,YP:
    PRINT Q$;:
    GET A$:
    XDRAW 66 AT XP,YP:
    A =  ASC (A$):
    N = N +  FN S(NN):
    ON A$ < >"Q" GOTO 56:
    RETURN
```

Place symbol at element centroid, display element number and move to another element.

```
58  VTAB 21:
    PRINT "ELEMENT #"N"   ":
    XP =  FN XC(N):
    YP =  FN YC(N):
    XDRAW 66 AT XP,YP:
    PRINT Q$;:
    GET A$:
    PRINT :
    XDRAW 66 AT XP,YP:
    A =  ASC (A$):
    N = N +  FN S(NE):
    ON A$ < >"Q" GOTO 58:
    RETURN
```

(60-62) Place a symbol on the diagonal between two adjacent elements, select diagonal to be reversed and move to another.

```
60  HOME :
    E1 = N:
    E2 = N +  1:
    PRINT "ELEMENTS: "E1" AND "E2:
    XP = .5 * ( FN XC(E1) +  FN XC(E2)):
    YP = .5 * ( FN YC(E1) +  FN YC(E2)):
    XDRAW 67 AT XP,YP

62  PRINT S$:
    GET A$:
    A =  ASC (A$):
    XDRAW 67 AT XP,YP:
    ON A$ = "S" GOSUB 64:
    N = E1:
    N = N +  FN S(NE):
    N = N +  FN S(NE):
    & C(CN%):
    ON A$ < >"Q" GOTO 60:
    RETURN
```

(64-72) Identify new element when diagonal is reversed, remove original and plot new diagonal.

```
64  & C(CN%):
    DIM CN%(7,1):
    FOR I = 1 TO 3:
        CN%(I,0) = I:
        CN%(I,1) = ND%(E1,I):
        J = I + 3:
        CN%(J,0) = J:
        CN%(J,1) = ND%(E2,I):
    NEXT I:
    K = 0:
    & O(CN%,1 TO 6,1,A):
    FOR I = 1 TO 6:
        A = CN%(I,1):
        N% = 1 + (A = CN%(I - 1,1)) * (I >1)
          + (A = CN%(I + 1,1)) * (I <6)

66      ON N% GOSUB 70,72:
    NEXT I:
    N1 = ND%(E1,CN%(7,0)):
    N2 = ND%(E2,CN%(7,1)):
    ND%(E1,CN%(7,0)) = CN%(0,1):
    ND%(E2,CN%(7,1)) = CN%(0,0)

68  X1 = FN PX(XN(N1)):
    Y1 = FN PY(YN(N1)):
    X2 = FN PX(XN(N2)):
    Y2 = FN PY(YN(N2)):
    HCOLOR= 0:
    HPLOT X1,Y1 TO X2,Y2 TO X1,Y1:
    N1 = CN%(0,0):
    N2 = CN%(0,1):
    X1 = FN PX(XN(N1)):
    Y1 = FN PY(YN(N1)):
    X2 = FN PX(XN(N2)):
    Y2 = FN PY(YN(N2)):
    HCOLOR= 3:
    HPLOT X1,Y1 TO X2,Y2 TO X1,Y1:
    RETURN

70  CN%(0,(CN%(I,0) >3)) = A:
    RETURN

72  ON (K >0 AND CN%(I,0) <4) OR
       (K <1 AND CN%(I,0) >3) OR
       (K = 1 AND A = ND%(E1,CN%(7,0))) GOTO 20:
    CN%(7,K) = CN%(I,0) - 3 * (CN%(I,0) >3):
    K = K + 1:
    RETURN
```

(74-134) Generate mesh, modify and save.

```
74  RESTORE :
    DATA -1,1,1,-1,1,-1,-1,1,1,-1,-1,1,-1,1,1,-1:
    FOR I = 1 TO 4:
        FOR J = 1 TO 4:
            READ IC(I,J):
    NEXT J,I:
    NN = 0:
    NE = 0:
    BW% = 0:
    FOR KK = 1 TO NQ:
        TEXT :
```

```
        & D(T):
        HOME :
        VTAB 1 +  20 * (PO >2):
        PRINT "     * GRID FOR INPUT REGION "KK" *[M]":
        IF PO >2 THEN
             GOSUB 198:
            HGR :
            & D(G)

76      POKE 34,2 +  19 * (PO >2):
        HOME :
        PRINT "[M]GENERATING NODAL COORDINATES";:
        FOR I = 1 TO 8:
            II = GN%(KK,I):
            XR(I) = X(II):
            YR(I) = Y(II):
        NEXT I:
        XR(9) = XR(1):
        YR(9) = YR(1):
        NR = NR%(KK):
        NC = NC%(KK):
        DA = 2 / (NR - 1):
        DS = 2 / (NC - 1)

78      FOR I = 1 TO NR:
             ET = 1 - (I - 1) * DA:
             FOR J = 1 TO NC:
                 SI =  - 1 +  (J - 1) * DS:
                 SH(1) =  - 0.25 * (1 - SI) * (1 - ET)
                     * (SI +  ET +  1):
                 SH(2) = .5 * (1 - SI ^ 2) * (1 - ET):
                 SH(3) = .25 * (1 +  SI) * (1 - ET)
                     * (SI - ET - 1):
                 SH(4) = .5 * (1 +  SI) * (1 - ET ^ 2):
                 SH(5) = .25 * (1 +  SI) * (1 +  ET)
                     * (SI +  ET - 1):
                 SH(6) = .5 * (1 - SI ^ 2) * (1 +  ET)

80               PRINT ".";:
                 SH(7) = .25 * (1 - SI) * (1 +  ET)
                     * (ET - SI - 1):
                 SH(8) = .5 * (1 - SI) * (1 - ET ^ 2):
                 XC(I,J) = 0:
                 YC(I,J) = 0:
                 FOR K = 1 TO 8:
                     XC(I,J) = XC(I,J) +  XR(K) * SH(K):
                     YC(I,J) = YC(I,J) +  YR(K) * SH(K):
        NEXT K,J,I:
        HOME

82      N1 = 1:
        S1 = 1:
        N2 = NR:
        S2 = NC:
        FOR I = 1 TO 4:
            ON JT%(KK,I) = 0
                 OR JT%(KK,I) >KK GOTO 96:
            NS = 0:
            FOR J = 1 TO 4:
                NS = NS +  J * (GN%(KK,2 * I)
                            = GN%(JT%(KK,I),2 * J)):
            NEXT J:
            K = NC * (I = 1 OR I = 3) +  NR
                * (I = 2 OR I = 4):
```

```
              JK = IC(I,NS):
              JL = (JK <> - 1) + K * (JK = - 1)

84            FOR J = 1 TO K:
                  ON I GOTO 86,88,90,92

86                NN%(NR,J) = RB%(JT%(KK,I),NS,JL):
                  N2 = NR - 1:
                  GOTO 94

88                NN%(J,NC) = RB%(JT%(KK,I),NS,JL):
                  S2 = NC - 1:
                  GOTO 94

90                NN%(1,J) = RB%(JT%(KK,I),NS,JL):
                  N1 = 2:
                  GOTO 94

92                NN%(J,1) = RB%(JT%(KK,I),NS,JL):
                  S1 = 2

94                JL = JL + JK:
              NEXT J

96        NEXT I:
          HOME :
          ON N1 >N2 OR S1 >S2 GOTO 98:
          FOR I = N1 TO N2:
              FOR J = S1 TO S2:
                  NN = NN + 1:
                  NN%(I,J) = NN:
          NEXT J,I:
          FOR I = 1 TO NC:
              RB%(KK,1,I) = NN%(NR,I):
              RB%(KK,3,I) = NN%(1,I):
          NEXT I:
          FOR I = 1 TO NR:
              RB%(KK,2,I) = NN%(I,NC):
              RB%(KK,4,I) = NN%(I,1):
          NEXT I

98        IF PO = 2 THEN
               PRINT "REGION "KK" NODE NUMBERS -":
              & L(P%):
              FOR I = 1 TO NR:
                  PRINT "   FOR ROW #"I":     ";:
                  FOR J = 1 TO NC:
                      PRINT NN%(I,J)"    ";:
                      NEXT J:
                      PRINT :
                  NEXT I:
                  & L(255):
                  GOSUB 4:
                  ON BC% = 2 GOTO 20

100       K = 1:
          FOR I = 1 TO NR:
              FOR J = 1 TO NC:
                  XE(K) = XC(I,J):
                  YE(K) = YC(I,J):
                  NE%(K) = NN%(I,J):
                  K = K + 1:
          NEXT J,I:
          IF PO = 4 THEN
```

```
          XA = 1:
          YA = 0:
          FOR I = 1 TO NR:
              FOR J = 1 TO NC:
                  L$ = STR$ (NN%(I,J)):
                  XP = FN PX(XC(I,J)):
                  YP = FN PY(YC(I,J)):
                  GOSUB 158:
          NEXT J,I

102   IF PO <3 THEN
          PRINT "[M]GENERATING ELEMENTS";:
          POKE 34,3

104   & C(B$):
      DIM B$(1):
      B$(0) = ".":
      & L(P% * (PO = 2)):
      FOR I = 1 TO NR - 1:
          FOR J = 2 TO NC:
              D1 = SQR ((XC(I,J) - XC(I
                 + 1,J - 1)) ^ 2 + (YC(I,J) - YC(I
                 + 1,J - 1)) ^ 2):
              D2 = SQR ((XC(I + 1,J) - XC(I,J - 1)) ^ 2
                 + (YC(I + 1,J) - YC(I,J - 1)) ^ 2):
              R(2) = NC * I + J:
              R(1) = R(2) - 1:
              R(3) = NC * (I - 1) + J:
              R(4) = R(3) - 1

106           FOR IJ = 1 TO 2:
                  NE = NE + 1:
                  B$(1) = "[M]ELEMENT #"
                     + STR$ (NE) + "[M]":
                  ON D1 / D2 >1.02 GOTO 108:
                  J1 = R(1):
                  J2 = R(IJ + 1):
                  J3 = R(IJ + 2):
                  GOTO 110

108               J1 = R(IJ):
                  J2 = R(IJ + 1):
                  J3 = R(4)

110               LB(0) = 0:
                  LB(1) = NE%(J1):
                  LB(2) = NE%(J2):
                  LB(3) = NE%(J3):
                  & O(LB,0 TO 3,0,A):
                  W% = LB(3) - LB(1) + 1:
                  BW% = BW% * (BW% > = W%)
                     + W% * (W% >BW%):
                  IF PO = 1 OR PO = 2 THEN
                      PRINT B$(PO = 2);

112               FOR IK = 1 TO 3:
                      J0 = J1 * (IK = 1) + J2
                         * (IK = 2) + J3 * (IK = 3):
                      ND%(NE,IK) = NE%(J0):
                      XN(ND%(NE,IK)) = XE(J0):
                      YN(ND%(NE,IK)) = YE(J0):
                      IF PO = 2 THEN
                          PRINT "  NODE "NE%(J0)
```

```
                              ": "DF$(1)"="XE(J0)"   "
                              DF$(2)"="YE(J0)

114                   NEXT IK:
                      ND%(NE,0) = KK:
                      ND%(NE,4) = NE:
                      N = NE:
                      ON PO >2 GOSUB 6:
                      IF PO = 5 THEN
                          L$ = "(" +   STR$ ( INT (NE
                             +  .1)) +  ")":
                          XP =  FN XC(NE):
                          YP =  FN YC(NE):
                          XA = 0:
                          YA = 0:
                          GOSUB 158

116       NEXT IJ,J,I:
          IF PO = 1 THEN
              HOME

118       ON PO < >1 GOSUB 4:
      NEXT KK:
      & L(255):
      & C(IC,XR,YR,SH,XC,YC,RB%,NN%,XE,YE,NE%,R,LB):
      GOSUB 146:
      GOSUB 144:
      ON NQ >1 GOSUB 188:
      TEXT :
      & D(T):
      HOME :
      PRINT "[M]PREPARING DATA FOR DISKETTE[M]":
      ON FR GOSUB 268:
      HOME :
      GOSUB 120:
      GOSUB 126:
      GOTO 132
```

(120-124) Save element node numbers.

```
120   A = (NE = 0):
      & B(A * 50,50):
      ON A GOTO 20:
      & C(EN%):
      DIM EN%(NE,3):
      FOR I = 1 TO NE:
          FOR J = 0 TO 3:
              EN%(I,J) = ND%(I,J):
      NEXT J,I:
      ON BW% = 0 GOSUB 134:
      EN%(0,0) = BW%:
      RE = 3:
      GOSUB 210:
      GOSUB 202:
      N1 = NE:
      N2 = 3:
      R1 = 5:
      ONERR  GOTO 226

122   EF = 0:
      GOSUB 204:
      ON  PEEK (2163) = 3 GOTO 124:
      & S(EN%,N$)
```

```
124 ON EF GOTO 122:
    POKE 216,0:
    GOSUB 206:
    GOSUB 224:
    & C(EN%):
    RETURN
```

(126-130) Save nodal coordinates.

```
126 A = (NN = 0):
    & B(A * 50,50):
    ON A GOTO 20:
    GOSUB 146:
    RE = 4:
    GOSUB 210:
    GOSUB 202:
    N1 = NN:
    N2 = 2:
    R1 = 5:
    ONERR  GOTO 226

128 EF = 0:
    GOSUB 204:
    ON  PEEK (2163) = 3 GOTO 130:
    & S(NC,N$)

130 ON EF GOTO 128:
    POKE 216,0:
    GOSUB 206:
    GOSUB 224:
    & C(NC):
    RETURN

132 PRINT "[M][M][M]SUMMARY OF GRID[M][M]"NE" ELEMENTS[M]"
      NN" NODES[M]BANDWIDTH="BW%"[M][M]":
    TEXT :
    & D(T):
    FR = FR * (Z < >3):
    GOTO 4

134 PRINT "[M][M]DETERMINING NEW BANDWIDTH...":
    & C(NE%,LB):
    DIM NE%(3),LB(1):
    FOR I = 1 TO NE:
        FOR J = 1 TO 3:
            NE%(J) = ND%(I,J):
        NEXT J:
        & O(NE%,0 TO 3,0,A):
        LB(0) = NE%(3) - NE%(1):
        & O(LB,0 TO 1,0,A):
    NEXT I:
    BW% = LB(1) + 1:
    PRINT " BW="BW%:
    & C(NE%,LB):
    RETURN
```

(136-146) Retrieve mesh information (RE=3,5 and 4,6).

```
136 RE = 3 + 2 * (RN = 1):
    GOSUB 210:
    GOSUB 202:
    ON SI%(RE) = 0 GOTO 156:
    ONERR  GOTO 226

138 EF = 0:
```

```
        GOSUB 214:
        ON BC% GOTO 20,20,136:
        NE = N1:
        ME = NE *  NOT RD +  ME * RD:
        & C(EN%):
        DIM EN%(NE,3):
        & R(EN%,N$):
        ON EF GOTO 138:
        GOSUB 222:
        RE = 4 +  2 * (RN = 1):
        GOSUB 202:
        ON SI%(RE) = 0 GOTO 156

140     EF = 0:
        GOSUB 214:
        ON BC% GOTO 136,20,136:
        NN = N1:
        MN = NN *  NOT RD +  MN * RD:
        & C(NC):
        DIM NC(NN,2):
        & R(NC,N$):
        ON EF GOTO 140:
        POKE 216,0:
        GOSUB 222

142     & C(XN,YN,ND%):
        DIM XN(MN),YN(MN),ND%(ME,4):
        BW% = EN%(0,0):
        NN = NN * (NN < = MN) +  MN * (NN >MN):
        FOR I = 1 TO NN:
             XN(I) = NC(I,1):
             YN(I) = NC(I,2):
        NEXT I:
        NE = NE * (NE < = ME) +  ME * (NE >ME):
        FOR I = 0 TO NE:
             ND%(I,4) = I:
             FOR J = 0 TO 3:
                  ND%(I,J) = EN%(I,J):
        NEXT J,I

144     & O(NC,1 TO NN,1,A):
        UL = NC(1,1):
        UH = NC(NN,1):
        & O(NC,1 TO NN,2,A):
        VL = NC(1,2):
        VH = NC(NN,2):
        & C(EN%,NC):
        RETURN
```

(146-154) Retrieve data from geometry program; region definition (RE=1,2).

```
146     & C(NC):
        DIM NC(NN,2):
        FOR I = 1 TO NN:
             NC(I,0) = I:
             NC(I,1) = XN(I):
             NC(I,2) = YN(I):
        NEXT I:
        RETURN

148     RE = 1:
        GOSUB 210:
        GOSUB 202:
        ON SI%(1) = 0 GOTO 156:
        ONERR  GOTO 226
```

```
150  EF = 0:
     GOSUB 214:
     ON BC% GOTO 20,20,148:
     NP = N1:
     & C(XY):
     DIM XY(NP,2):
     & R(XY,N$):
     ON EF GOTO 150:
     GOSUB 222:
     RE = 2:
     GOSUB 202:
     ON SI%(2) = 0 GOTO 156

152  EF = 0:
     GOSUB 214:
     NQ = N1:
     & C(RD%):
     DIM RD%(NQ,14):
     & R(RD%,N$):
     ON EF GOTO 152:
     POKE 216,0:
     GOSUB 222:
     & C(X,Y,JT%,NR%,NC%,GN%)

154  DIM X(NP),Y(NP),JT%(NQ,4),NR%(NQ),NC%(NQ),GN%(NQ,8):
     FOR I = 1 TO NP:
          X(I) = XY(I,1):
          Y(I) = XY(I,2):
     NEXT I:
     FOR I = 1 TO NQ:
          FOR J = 1 TO 4:
               JT%(I,J) = RD%(I,J):
          NEXT J:
          NR%(I) = RD%(I,5):
          NC%(I) = RD%(I,6):
          FOR J = 1 TO 8:
               GN%(I,J) = RD%(I,6 +  J):
     NEXT J,I:
     & C(XY,RD%):
     GOTO 4
```

Warn of invalid data.

```
156  FLASH :
     PRINT "ERROR # "; PEEK (222);"[M]":
     ON  PEEK (222) = 77 GOTO 232:
     PRINT "[G][G][M]DATA NOT DEFINED":
     NORMAL :
     PRINT "EITHER FILE":
     INVERSE :
     PRINT NA$:
     NORMAL :
     PRINT "DOESN'T EXIST OR THE DATA IS NOT VALID":
     GOSUB 4:
     BC% = 2:
     RETURN
```

Draw L$ label on graphics screen.

```
158  FOR I1 = 1 TO  LEN (L$):
          DRAW  ASC ( MID$ (L$,I1,1)) - 31 AT XP - 7
             *  LEN (L$) / 2 +  7 * (I1 - 1) +  XA * (3.5
             *  LEN (L$) +  5),YP +  3.5 - YA * 5:
     NEXT I1:
     RETURN
```

(160-162) List nodal coordinates and element node numbers in counterclockwise order.

```
160  FR = 0:
     HOME :
     & L(P%):
     PRINT  SPC( (40 -  LEN (HE$)) / 2)HE$"[M]":
     INVERSE :
     PRINT "NODE     "DF$(1) TAB( 25)DF$(2) TAB( 39)" ";
     NORMAL :
     POKE 34,5:
     VTAB 6:
     FOR I = 1 TO NN:
        PRINT "  "I TAB( 10)XN(I) TAB( 25)YN(I):
     NEXT I:
     POKE 34,2:
     GOSUB 4

162  PRINT  SPC( (40 -  LEN (HE$)) / 2)HE$"[M]":
     INVERSE :
     PRINT "ELEMENT     NODE NUMBERS":
     NORMAL :
     POKE 34,5:
     VTAB 6:
     FOR I = 1 TO NE:
        PRINT "   "I;:
        FOR J = 1 TO 3:
           PRINT  TAB( 7 +  5 * J)ND%(I,J);:
        NEXT J:
        PRINT :
     NEXT I:
     POKE 34,0:
     GOSUB 4:
     PRINT "[M]*** BANDWIDTH IS: "BW%:
     & L(255):
     GOTO 4
```

(164-166) List nodes of unique lines, boundary lines, and nodes in counterclockwise order.

```
164  FR = 0:
     HOME :
     & L(P%):
     PRINT  SPC( (40 -  LEN (HE$))
        / 2)HE$"[M][M]     UNIQUE GRID LINES[M]":
     INVERSE :
     PRINT "NODE NUMBERS    BOUNDARY":
     NORMAL :
     POKE 34,7:
     VTAB 8:
     FOR I = 1 TO NL:
        PRINT "   "AN%(I,1) TAB( 8)AN%(I,2);:
        IF AN%(I,3) >0 THEN
           PRINT  TAB( 17)"YES";

166     PRINT :
     NEXT I:
     POKE 34,2:
     GOSUB 4:
     VTAB 3:
     INVERSE :
     PRINT "BOUNDARY  RELATED[M] NODE     ELEMENT":
     NORMAL :
     PRINT :
     POKE 34,5:
     VTAB 6:
```

```
        FOR I = 1 TO NB:
            PRINT  TAB( 5)BN%(I,1) TAB( 14)BN%(I,2):
        NEXT I:
        & L(255):
        POKE 34,0:
        GOTO 4
```

(168-178) Define table of related nodes and renumber to reduce bandwidth.

```
168  & C(N%,RN%,NW%,OL%):
        DIM N%(NN),RN%(NN,8),NW%(NN),OL%(NN):
        HOME :
        PRINT "FINDING 'RELATED NODES' FOR EACH NODE":
        POKE 34, PEEK (37):
        GOSUB 22:
        POKE 34,2:
        HOME :
        & B(30,10)

170  IF PO = 3 THEN
            & L(P%):
            PRINT "NODE        RELATED NODES":
            POKE 34,3:
            FOR I = 1 TO NN:
                PRINT " "I;:
                FOR J = 1 TO N%(I):
                    PRINT  TAB( 5 +  4 * J)RN%(I,J);:
                NEXT J:
                PRINT :
            NEXT I:
            & L(255):
            GOSUB 4

172  TEXT :
        & D(T):
        HOME :
        PRINT "        * RE-ASSIGNING NODE NUMBERS *[M]
           [M]     (ORIGINAL BANDWIDTH = "BW%")[M][M]":
        POKE 34,4:
        BI = 0:
        I1 = 1:
        I2 = NN:
        GOSUB 30:
        ON BI >0 GOTO 176:
        HOME :
        PRINT "[M]* NO REDUCTION IN BANDWIDTH --[M]
           ORIGINAL NODE NUMBERING USED *"

174  POKE 34,0:
        GOSUB 210:
        RE = 3:
        GOSUB 202:
        RE = 5:
        GOSUB 206:
        RE = 4:
        GOSUB 202:
        RE = 6:
        GOSUB 206:
        GOTO 4

176  POKE 34,4:
        HOME :
        PRINT "[M]MINIMUM BANDWIDTH (BW="BW%") OCCURRED[M]
           WHEN NUMBERING STARTED AT NODE #"BI".[M][M]
           RENUMBERING WITH THIS STARTING POINT";:
```

```
    P0 = PO:
    PO = 0:
    I1 = BI:
    I2 = BI:
    POKE 34,0

178 GOSUB 30:
    PO = P0:
    IF PO >1 THEN
        HOME :
        & L(P%):
        PRINT "** RENUMBERED NODES **[M][M]OLD #    NEW #":
        POKE 34,3:
        FOR I = 1 TO NN:
            PRINT "   "I TAB( 12)NW%(I):
        NEXT I:
        & L(255):
        POKE 34,0:
        GOSUB 4
```

(180-186) Save renumbered nodes and nodal coordinates.

```
180 & C(NC,EN%):
    DIM NC(NN,2),EN%(NE,3):
    FOR I = 1 TO NN:
        NC(I,1) = XN(OL%(I)):
        NC(I,2) = YN(OL%(I)):
    NEXT I:
    & O(ND%,1 TO NE,4,A):
    FOR EL = 1 TO NE:
        EN%(EL,0) = ND%(EL,0):
        FOR J = 1 TO 3:
            EN%(EL,J) = NW%(ND%(EL,J)):
    NEXT J,EL

182 EN%(0,0) = BW%:
    RE = 5:
    GOSUB 210:
    GOSUB 202:
    N1 = NE:
    N2 = 3:
    R1 = 7:
    ONERR  GOTO 226

184 GOSUB 122:
    RE = 6:
    GOSUB 202:
    N1 = NN:
    N2 = 2:
    ONERR  GOTO 226

186 GOSUB 128:
    & C(N%,RN%,NW%,OL%,XN,YN,ND%):
    FOR I = 3 TO 6:
        SI%(I) = 0:
    NEXT I:
    GOTO 4
```

Plot composite mesh, including all regions.

```
188 VTAB 22:
    HGR :
    & D(G):
    POKE 34,20:
    HOME :
    PRINT  TAB( 14)"DEFINED GRID":
```

```
     GOSUB 200:
     FOR N = 1 TO NE:
        GOSUB 6:
     NEXT N:
     GOTO 4
```

(190-196) Place cursor on each node or element centroid to reveal renumbering.

```
190  TEXT :
     & D(T):
     HOME :
     PRINT "IDENTIFY NODES (2.71)[M]":
     & I("START AT # ";1;N,"4",
        BC%,0,N >0 AND N < = NN):
     ON BC% GOTO 20,194,190:
     PRINT "[M]"M$"NODE IDENTIFICATION.":
     GOSUB 4

192  ON BC% GOTO 190,194,190:
     VTAB 21:
     POKE 34,20:
     PRINT "NODE #"N"[M]"DF$(1)"="XN(N)"[M]"
        DF$(2)"="YN(N):
     & D(G):
     GOSUB 56

194  TEXT :
     & D(T):
     HOME :
     PRINT "IDENTIFY ELEMENTS (2.72)[M]":
     & I("START AT # ";1;N,"5",
        BC%,0,N >0 AND N < = NE):
     ON BC% GOTO 190,20,194:
     PRINT "[M]"M$"ELEMENT IDENTIFICATION":
     GOSUB 4

196  ON BC% GOTO 194,20,196:
     POKE 34,20:
     HOME :
     & D(G):
     GOSUB 58:
     HOME :
     PRINT  TAB( 14)"DEFINED GRID":
     GOSUB 4:
     ON BC% GOTO 190:
     RETURN
```

Find range of coordinates for points in this region.

```
198  & C(XY):
     DIM XY(8,2):
     FOR I = 1 TO 8:
         J = GN%(KK,I):
         XY(I,1) = X(J):
         XY(I,2) = Y(J):
     NEXT I:
     & O(XY,1 TO 8,1,A):
     UL = XY(1,1):
     UH = XY(8,1):
     & O(XY,1 TO 8,2,A):
     VL = XY(1,2):
     VH = XY(8,2):
     & C(XY)
```

Define plotting scale and clear graphics screen.

```
200  S0 = (SB - ST) / (VH - VL):
```

```
     SC = (SR - SL) / (UH - UL):
     SC = SC * (SC < = S0) +  S0 * (SC >S0):
     RETURN
```

Read file information for record RE.

```
202  N =  FRE (0):
     PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"READ"FI$",R"RE:
     INPUT SI%(RE),NA$,DE$,N1,D1$,N2,D2$:
     PRINT D$"CLOSE"FI$:
     GOTO 218
```

Request file name for saving data.

```
204  ON  PEEK (2163) > = 3 GOTO 218:
     PRINT "[M]*** ";:
     INVERSE :
     PRINT "SAVING";:
     NORMAL :
     PRINT " DATA TO DISKETTE ***[M]":
     ON FR GOTO 218:
     ON  PEEK (2163) GOTO 220:
     & I("DATA FILE NAME FOR[M]"
       +  DE$,NA$,NA$,"6",BC%,30):
     ON BC% >0 GOTO 204:
     GOTO 216
```

Set file status to active, save information about record RE, and print description.

```
206  SI%(RE) = 1:
     ON  PEEK (2163) = 3 GOTO 20:
     PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"WRITE"FI$",R"RE:
     PRINT 1:
     PRINT NA$:
     PRINT DE$:
     PRINT N1:
     PRINT D1$:
     PRINT N2:
     PRINT D2$:
     PRINT D$"CLOSE"FI$

208  PRINT "[M]"DE$"[M]FOR "N1" "D1$" & "N2" "D2$"[M]
       WERE SAVED":
     ON FR = 0 GOTO 20:
     PRINT "IN FILE: "NA$:
     RETURN
```

(210-212) Verify existence of data file.

```
210  ONERR  GOTO 238

212  E0 = 0:
     PRINT D$:
     PRINT D$"CKFILE"FI$;DR$(DR):
     ON E0 GOTO 212:
     POKE 216,0:
     RETURN
```

Request data retrieval filename.

```
214  ON Z = 2 GOTO 218:
     PRINT "[M]*** ";:
     INVERSE :
     PRINT "LOADING";:
     NORMAL :
     PRINT " DATA FROM DISKETTE ***[M]":
```

```
     ON FR GOTO 218:
     ON  PEEK (2163) GOTO 220:
     & I("DATA FILE NAME FOR[M]"
        +  DE$,NA$,NA$,"7",BC%,30):
     ON BC% >0 GOTO 214
```

Assure presence of keyword.

```
216  ON  LEFT$ (NA$, LEN (KW$)) = KW$ GOTO 218:
     NA$ = KW$ +  "/" +  NA$

218  N$ = NA$ +  DR$(DR):
     RETURN

220  PRINT " "NA$:
     GOTO 218
```

Print data descriptors.

```
222  ON Z = 2 GOTO 20:
     PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2" "D2$"[M]
        WERE LOADED":
     ON FR = 0 GOTO 20:
     PRINT " FROM FILE: "NA$:
     RETURN
```

Disable data files.

```
224  ON  PEEK (2163) = 3 GOTO 20:
     PRINT D$"OPEN"FI$",L100"DR$(DR):
     FOR RE = R1 TO 22:
          PRINT D$"WRITE"FI$",R"RE:
          PRINT 0:
          SI%(RE) = 0:
     NEXT RE:
     PRINT D$"CLOSE"FI$:
     RETURN
```

(226-238) Trap errors.

```
226  EF = 1:
     ER =  PEEK (222):
     POKE 216,0:
     ON ER = 255 GOTO 230:
     FLASH :
     ON ER = 77 GOTO 232:
     PRINT "[G][G][M]ERROR # "ER"[M]":
     NORMAL :
     ER = ER * (ER >3 AND ER <11)
        +  NOT (ER >3 AND ER <11):
     ON ER = 1 OR ER = 5 GOTO 228:
     ON ER >3 AND ER <11 GOTO 234

228  GOSUB 202:
     RESUME

230  END

232  PRINT "[G][G]SORRY, OUT OF MEMORY":
     END

234  PRINT ER$(ER - 4):
     ON ER = 6 GOSUB 236:
     ON ER = 4 OR ER = 5 OR ER = 8 OR ER = 9 GOSUB 4:
     ON ER <10 GOTO 228:
     & I("UNLOCK? (Y/N)";"Y";A$,"8",0,1):
```

```
      ON A$ = "N" GOTO 228:
      PRINT D$"UNLOCK "NA$DR$(DR):
      GOTO 228

 236  & I("[M]CATALOG? (Y/N)";"N";A$,"9",0,1):
      ON A$ < >"Y" GOTO 20:
      PRINT D$"CATALOG"DR$(DR):
      RETURN

 238  E0 = 1:
      POKE 216,0:
      PRINT "[G][G][M]DATA DISKETTE IS NOT IN DRIVE":
      GOSUB 4:
      RESUME
```

(1,240-254) Cold start initialization.

```
 240  TEXT :
      & D(T):
      HOME :
      P% = 10:
      INVERSE :
      FOR I = 1 TO 4:
         PRINT  TAB( 2)" " TAB( 38)" ":
      NEXT I:
      NORMAL :
      VTAB 2:
      HTAB 3:
      PRINT " APPLIED FINITE ELEMENT ANALYSIS: ":
      HTAB 3:
      PRINT "    AN APPLE "
         CHR$ (221) CHR$ (219)" IMPLEMENTATION    ":
      VTAB 6:
      HTAB 12:
      PRINT "COPYRIGHT, 1985[M]"

 242  HTAB 14:
      INVERSE :
      PRINT "GRID";:
      NORMAL :
      PRINT "  (2.00)[M][M][M]ABSTRACT:[M][M]
         A GRID OF  3-NODE TRIANGULAR ELEMENTS
         ISGENERATED FOR THE PROBLEM DOMAIN.  NEXT,
         NODES ARE RENUMBERED  TO REDUCE  PROBLEMBANDWIDTH.
         THEN UNIQUE LINES AND  NODESON";

 244  PRINT " BOUNDARIES ARE IDENTIFIED.":
      GOSUB 4:
      DIM ER$(6),DR$(2),SI%(22):
      ER$(0) = "DISKETTE IS WRITE PROTECTED":
      ER$(2) = "FILE NOT FOUND":
      ER$(3) = "VOLUME MISMATCH":
      ER$(4) = "I/O ERROR - DOOR OPEN OR DISK INIT":
      ER$(5) = "DISK FULL":
      ER$(6) = "FILE LOCKED"

 246  DE = 3:
      HCOLOR= 3:
      SCALE= 1:
      ROT= 0:
      FI$ = "FILEINFO.TXT":
      FR = 0:
      DEF  FN XC(N) =  FN PX((XN(ND%(N,1)) + XN(ND%(N,2))
         + XN(ND%(N,3))) / 3):
```

```
     DEF  FN YC(N) =  FN PY((YN(ND%(N,1)) +  YN(ND%(N,2))
        +  YN(ND%(N,3))) / 3)

248  ST = 10:
     SB = 150:
     SL = 20:
     SR = 250:
     DEF  FN PX(X) = SL +  SC * (X - UL):
     DEF  FN PY(Y) = SB - SC * (Y - VL):
     DR$(1) = ",D1":
     DR$(2) = ",D2":
     DR =  PEEK (2048):
     IF DR = 1 THEN
          & B(200,20):
        PRINT "INSERT ";:
        FLASH :
        PRINT "DATA";:
        NORMAL :
        PRINT " DISKETTE INTO DRIVE":
        GOSUB 4

250  FR = ( PEEK (2163) = 3):
     GOSUB 210:
     PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"READ"FI$",R0":
     INPUT TE,KW$,PD$:
     PRINT D$"CLOSE"FI$:
     PRINT "[M]PROBLEM KEYWORD: "KW$"[M][M]
        PROBLEM DESCRIPTION:[M]"PD$:
     DIM MU$(7):
     MU$(1) = "DO COMPLETE GRID GENERATION"

252  MU$(2) = "DEFINE GRID ELEMENTS & NODES":
     MU$(3) = "RENUMBER NODES TO REDUCE BANDWIDTH":
     MU$(4) = "SORT UNIQUE LINES & BOUNDARY NODES":
     MU$(5) = "LIST ELEMENT NODAL COORDINATES":
     MU$(6) = "LIST LINE & BOUNDARY NODES":
     MU$(7) = "PLOT ENTIRE GRID"

254  GOSUB 220:
     GOSUB 4:
     DIM DI$(4),DF$(2):
     DI$(1) = "X":
     DI$(2) = "Y":
     DI$(3) = "R":
     DI$(4) = "Z":
     DF$(1) = DI$(2 * TE - 1):
     DF$(2) = DI$(2 * TE):
     DEF  FN P2(I) = I +  1 - DE * (I = DE):
     DEF  FN P1(I) = I - 1 +  DE * (I = 1):
     DEF  FN S(NE) = (A = 21) * (1 - NE * (N = NE))
        +  (A = 8) * ( - 1 +  NE * (N = 1))
```

(2,256-260) Restart; menu.

```
256  RD = 0:
     POKE 216,0:
     NM =  PEEK (2166):
     TEXT :
     & D(T):
     HOME :
     EO = 1:
     ON Z = 2 AND  PEEK (2165) = 3 GOTO 264:
     PRINT "[M]  ** GRID  (2.05) **[M][M]
```

```
      -- GRID DEFINITION --[M]
      1. "MU$(1)" (2-4)[M][M]
      2. "MU$(2)"[M]
      3. "MU$(3)"[M]
      4.  "MU$(4)"[M][M]
      -- GRID EXAMINATION --"

258  PRINT "5. "MU$(5)"[M]
      6. "MU$(6)"[M]
      7. "MU$(7)"[M][M]
      0.NONE OF THE ABOVE":
     MO = 1:
     ON Z = 2 GOSUB 268:
     ON Z = 2 GOTO 260:
     & I(SR$,NM;MO,"10",BC%,1,
        MO > = 0 AND MO <8):
     ON BC% >0 GOTO 262

260  ON MO = 0 GOTO 262:
     HOME :
     PRINT MU$(MO)" (2."MO")":
     POKE 34,2:
     ON MO GOSUB 270,274,288,294,308,310,318:
     GOTO 256
```

(262-268) Exit menu.

```
262  HOME :
     PRINT "[M]    ** GRID: EXIT (2.8) **[M][M]
      1. PROCEED TO 'PREPROCESS'[M]
      2. EXIT TO PROGRAM MENU[M]
      3. DON'T EXIT: REMAIN IN 'GRID'[M][M]
      0. STOP[M]":
     & I(SR$,1;EO,"11",BC%,1,
        EO > = 0 AND EO <4):
     ON BC% >0 GOTO 262:
     ON EO = 3 GOTO 256:
     & C(PN$):
     ON EO = 0 GOTO 230

264  DIM PN$(1):
     PN$(0) = "HELLO":
     PN$(1) = "PREPROCESS":
     IF DR = 1 THEN
         & B(200,20):
        PRINT "[M]INSERT ";:
        FLASH :
        PRINT "PROGRAM";:
        NORMAL :
        PRINT " DISKETTE CONTAINING[M]'";:
        INVERSE :
        PRINT PN$(EO = 1);:
        NORMAL :
        PRINT "' INTO DRIVE":
        FR = 0:
        GOSUB 4:
        ON BC% = 2 OR BC% = 3 GOTO 262

266  POKE 2166,1:
     PRINT D$"RUN "PN$(EO = 1)",D1"

268  & B(10,10):
     & B(0,0,12):
     RETURN
```

270-272) Menu option 1: Do complete mesh generation.

```
270  MO = 1:
     ON Z = 2 GOTO 272:
     PRINT "[M]RUN OPTIONS:[M][M]
        1. W/O USER INTERACTION[M]    (USING ALL DEFAULTS)[M]
        2. WITH USER INTERACTION":
     & I(SR$,1;FR,"12",BC%,1,FR = 1 OR FR = 2):
     ON BC% >0 GOTO 20:
     FR = (FR = 1)

272  FR = 1:
     HOME :
     MO = MO +  1:
     PRINT "DOING MENU OPTION "MO"-[M]   "MU$(MO)"[M]":
     GOSUB 268:
     ON MO - 1 GOSUB 278,288,294:
     ON MO <4 GOTO 272:
     ON FR = 0 GOTO 20:
     & B(50,20,3):
     RETURN
```

(274-286) Menu option 2: Define mesh elements and nodes.

```
274  POKE 34,2:
     HOME :
     & C(GM$,K$,N$):
     DIM GM$(3),K$(2),N$(2):
     GM$(1) = "GENERATE GRID":
     GM$(2) = "INPUT GRID FROM KEYBOARD":
     GM$(3) = "MODIFY EXISTING GRID":
     K$(1) = " COORDINATES OF NODES":
     K$(2) = " NODES OF ELEMENTS":
     N$(1) = " NODE":
     N$(2) = " ELEMENT"

276  PRINT "[M]1. "GM$(1)"[M]
        2. "GM$(2)"[M]
        3. "GM$(3)"[M][M]0. NONE[M]":
     & I(SR$,1;GM,"13",BC%,1,
        GM > = 0 AND GM < = 3):
     ON BC% >0 OR GM = 0 GOTO 20:
     HOME :
     PRINT "- "GM$(GM)" (2.2"GM") -[M]":
     POKE 34,3:
     ON GM GOTO 278,336,350

278  GM = 1:
     ON SI%(1) = 0 OR SI%(2) = 0 GOSUB 148:
     ON BC% = 2 GOTO 20:
     ON FR GOTO 280:
     HOME :
     PRINT "[M]DISPLAY OPTIONS[M][M]
        1. WITH PLOTTING[M]
        2. WITHOUT PLOTTING[M]":
     & I(SR$,1;SO,"14",BC%,1,SO >0 AND SO <3):
     ON BC% GOTO 20,278,278

280  SO = SO * (FR = 0) +  FR:
     ON SO = 1 GOTO 282:
     HOME :
     PRINT "[M]GRID LISTING OPTIONS -[M][M]
        1. SHOW ONLY CALCULATION STATUS[M]
        2. LIST NODES & COORDINATES OF ELEMENTS[M]":
```

```
     & I(SR$,1;PO,"15",BC%,1,PO >0 AND PO <3):
     ON BC% GOTO 278,280,280:
     GOTO 284

282  HOME :
     ON FR GOTO 284:
     PRINT "[M]GRID PLOTTING OPTIONS -[M][M]
        1. WITHOUT LABELS[M]
        2. WITH NODE LABELS[M]
        3. WITH ELEMENT LABELS[M]":
     & I(SR$,1;PO,"16",BC%,1,PO >0 AND PO <4):
     ON BC% GOTO 278,282,282

284  PO = PO * (FR = 0) +  2 * (SO = 1 AND FR = 0)
        +  3 * (FR = 1):
     MR = NR%(1):
     MC = NC%(1):
     ME = 0:
     MN = 0:
     FOR I = 1 TO NQ:
         N1 = NR%(I):
         N2 = NC%(I):
         MR = MR * (MR > = N1) +  N1 * (N1 >MR):
         MC = MC * (MC > = N2) +  N2 * (N2 >MC):
         MN = MN +  N1 * N2:
         ME = ME +  (N1 - 1) * (N2 - 1) * 2:
     NEXT I:
     MX = MR * (MR > = MC) +  MC * (MR <MC):
     A = MR * MC

286  & C(IC,XR,YR,SH,XC,YC,RB%):
     DIM IC(4,4),XR(9),YR(9),SH(8),XC(MR,MC),YC(MR,MC),
        RB%(NQ,4,MX):
     & C(NN%,XE,YE,NE%,R,LB,XN,YN,ND%):
     DIM NN%(MX,MX),XE(A),YE(A),NE%(A),R(4),LB(3),
        XN(MN),YN(MN),ND%(ME,4):
     GOSUB 74:
     POKE 2166,3:
     RETURN
```

(288-292) Menu option 3: Renumber nodes to reduce bandwidth.

```
288  RN = 0:
     PO = PO * (FR < >1) +  (FR = 1):
     GOSUB 136:
     ON BC% = 2 GOTO 20:
     GOSUB 4

290  ON FR GOTO 292:
     HOME :
     PRINT "[M]RENUMBERING DISPLAY[M][M]
        1. ONLY CALCULATION STATUS[M]
        2. STATUS AND FINAL RENUMBERING[M]
        3. DETAILS OF RENUMBERING[M]":
     & I(SR$,1;PO,"17",BC%,1,PO >0 AND PO <4):
     ON BC% GOTO 20,288,288

292  GOSUB 168:
     POKE 2166,4:
     RETURN
```

(294-306) Menu option 4: Sort unique lines and boundary nodes (RE=7,8).

```
294  RN = 1:
     PO = PO * (FR < >1) +  (FR = 1):
```

```
     IF SI%(5) = 0 THEN
          GOSUB 136:
          ON BC% = 2 GOTO 20:
          GOSUB 4

296  ON FR GOTO 298:
     HOME :
     PRINT "[M]UNIQUE LINE DISPLAY[M][M]
        1. ONLY CALCULATION STATUS[M]
        2. STATUS AND FINAL LISTS[M]":
     & I(SR$,1;PO,"18",BC%,1,PO >0 AND PO <3):
     ON BC% GOTO 20,294,294

298  POKE 2166,5:
     POKE 2165,3:
     & C(AN%,LN%):
     MA = 2 * NE +  9:
     DIM AN%(MA,3):
     GOSUB 40:
     DIM LN%(NL,3):
     FOR I = 1 TO NL:
          FOR J = 1 TO 3:
               LN%(I,J) = AN%(I,J):
     NEXT J,I:
     RE = 7:
     GOSUB 210:
     GOSUB 202:
     N1 = NL:
     N2 = 3:
     ONERR  GOTO 226

300  EF = 0:
     GOSUB 204:
     ON  PEEK (2163) = 3 GOTO 302:
     & S(LN%,N$)

302  ON EF GOTO 300:
     POKE 216,0:
     GOSUB 206:
     R1 = 9:
     GOSUB 224:
     RE = 8:
     GOSUB 202:
     N1 = NB:
     N2 = 2

304  EF = 0:
     GOSUB 204:
     ON  PEEK (2163) = 3 GOTO 306:
     & S(BN%,N$)

306  ON EF GOTO 304:
     POKE 216,0:
     GOSUB 206:
     GOSUB 4:
     RETURN
```

Menu option 5: List node numbers and coordinates (original and renumbered).

```
308  FR = 0:
     & I("[M]RENUMBERED NODES";"Y";A$,"19",0,1):
     RN = (A$ = "Y"):
     GOSUB 136:
     ON BC% = 2 GOTO 20:
     HE$ = "GRID ELEMENTS AND NODES":
```

```
         GOSUB 160:
         POKE 2166,6:
         RETURN

             (310-316) Menu option 6: List line and boundary nodes
310   FR = 0:
      ON SI%(7) GOTO 316:
      RE = 7:
      GOSUB 210:
      GOSUB 202:
      ON SI%(7) = 0 GOTO 156:
      ONERR  GOTO 226

312   EF = 0:
      GOSUB 214:
      ON BC% >0 GOTO 20:
      NL = N1:
      & C(AN%):
      DIM AN%(NL,3):
      & R(AN%,N$):
      ON EF GOTO 312:
      GOSUB 222:
      RE = 8:
      GOSUB 202:
      ON SI%(RE) = 0 GOTO 156

314   EF = 0:
      GOSUB 214:
      ON BC% GOTO 310,20,310:
      NB = N1:
      & C(BN%):
      DIM BN%(NB,2):
      & R(BN%,N$):
      ON EF GOTO 314:
      POKE 216,0:
      GOSUB 222:
      GOSUB 4

316   HE$ = "UNIQUE LINES AND BOUNDARY NODES":
      GOSUB 164:
      POKE 2166,7:
      RETURN
             Menu option 7: Plot entire mesh, and identify elements and nodes.
318   FR = 0:
      & I("[M]RENUMBERED NODES";"Y";A$,"20",0,1):
      RN = (A$ = "Y"):
      GOSUB 136:
      ON BC% = 2 GOTO 20:
      GOSUB 188:
      & I("[M]IDENTIFY NODES OR ELEMENTS";"N";A$,"21",0,1):
      ON (A$ = "Y") GOSUB 190:
      RETURN

             (320-352)Modify nodes and elements.
320   & I(DF$(1) +  "=";XN(N);XN(N),"22",BC%):
      ON BC% >0 GOTO 20:
      & I(DF$(2) +  "=";YN(N);YN(N),"23",BC%):
      ON BC% = 1 GOTO 320:
      RETURN

322   A = ND%(N,0):
      A = A * (A >0) +  (NQ +  1) * (A = 0):
      & I("[M]- PART OF INPUT REGION #";
        A;ND%(N,0),"24",BC%):
```

```
     ON BC% >0 GOTO 20:
     FOR J = 1 TO 3:
         & I(" NODE " +  STR$ (J)
           + ":";ND%(N,J);ND%(N,J),"25",BC%,0,
           ND%(N,J) > = 1 AND ND%(N,J) < = MN):
         J = J - (BC% >0) - (BC% = 1)
           * (J >1) +  3 * (BC% = 2):
     NEXT J
```

Check counterclockwise order of nodes in an element.

```
324  & C(XY):
     DIM XY(3,2):
     FOR J = 1 TO 3:
         XY(J,1) = XN(ND%(N,J)):
         XY(J,2) = YN(ND%(N,J)):
     NEXT J:
     A = 0:
     FOR J = 1 TO 3:
         A = A +  XY(J,1) * XY( FN P2(J),2)
            - XY( FN P2(J),1) * XY(J,2):
     NEXT J:
     ON A > - 1E - 8 GOTO 20:
     PRINT "[G]ENTER NODES IN CCW ORDER":
     GOSUB 4:
     PRINT "FOR "N$" #"N:
     GOTO 322

326  PO = PO * (FR = 0) +  2 * (SO = 1 AND FR = 0)
       +  3 * (FR = 1):
     MR = NR%(1):
     MC = NC%(1):
     ME = 0:
     MN = 0:
     FOR I = 1 TO NQ:
         N1 = NR%(I):
         N2 = NC%(I):
         MR = MR * (MR > = N1) +  N1 * (N1 >MR):
         MC = MC * (MC > = N2) +  N2 * (N2 >MC):
         MN = MN +  N1 * N2:
         ME = ME +  (N1 - 1) * (N2 - 1) * 2:
     NEXT I:
     MX = MR * (MR > = MC) +  MC * (MR <MC):
     A = MR * MC

328  & C(NC):
     DIM NC(NN,2):
     FOR I = 1 TO NN:
         NC(I,0) = I:
         NC(I,1) = XN(I):
         NC(I,2) = YN(I):
     NEXT I:
     RETURN
```

(330-332) Adjust position of an existing node.

```
330  POKE 34,3:
     HOME :
     PRINT "ADJUSTING NODE POSITIONS[M][M]INSTRUCTIONS:[M]"
        M$"NODE SELECTION.[M][M]
        USE <S>TO SELECT A NODE[M]
        - USE <I><M><J><K>
        TO MOVE NODE[M]    UP, DOWN, LEFT, RIGHT.[M]
        - USE <V>TO INPUT COORDINATE VALUES.[M]
        - USE <Q>TO QUIT";
```

```
332  PRINT "POSITION ADJUSTMENT.[M][M]CTRL-G FOR VIEWING PLOT[M]
        CTRL-T FOR VIEWING INSTRUCTIONS.":
     VTAB 21:
     GOSUB 4:
     ON BC% >0 GOTO 20:
     & D(G):
     VTAB 21:
     POKE 34,20:
     N = 1:
     GOSUB 10:
     ON A = 20 GOTO 330:
     RETURN
```

Reverse the diagonal between adjacent elements.

```
334  HOME :
     PRINT "REVERSING ELEMENT DIAGONALS[M][M]
        INSTRUCTIONS:[M]
        1. MOVE CURSOR USING LEFT/RIGHT ARROWS[M]
        2. <S<SELECTS ELEMENTS[M]
        3. <Q>STOPS ELEMENT CHANGES":
     GOSUB 4:
     ON BC% >0 GOTO 20:
     & D(G):
     VTAB 22:
     POKE 34,20:
     N = 1:
     NC = 4:
     GOTO 60
```

(336-348) Enter nodes and elements from the keyboard (without graphics).

```
336  HOME :
     & C(XN,YN,ND%):
     NN = 0:
     NE = 0:
     MN = 0:
     ME = 0:
     & I("[M]RETRIEVE EXISTING GRID (Y/N)";"N";A$,
        "26",BC%,1):
     ON BC% >0 GOTO 20:
     D1 = (A$ = "Y"):
     ON D1 = 0 GOTO 338:
     RN = 0:
     GOSUB 136:
     ON BC% >0 GOTO 20:
     GOSUB 188:
     POKE 34,3:
     & D(T):
     HOME :
     PRINT "[M][M]"NN" NODES AND "NE" ELEMENTS EXIST"

338  PRINT "[M]ALLOCATE SPACE FOR GRID -[M]":
     & I("- TOTAL NODES";MN;N1,"27",BC%):
     ON BC% GOTO 336,20,338:
     & I("- TOTAL ELEMENTS";ME;N2,"28",BC%):
     ON BC% GOTO 338,20,338

340  RD = MN < >N1 OR ME < >N2:
     MN = N1:
     ME = N2:
     ON RD AND D1 GOSUB 136:
     IF RD = 1 AND D1 = 0 THEN
           DIM XN(MN),YN(MN),ND%(ME,DE)

342  POKE 34,3:
```

```
          HOME :
          PRINT "[M]1. DEFINE"K$(1)"[M]
             2. DEFINE"K$(2)"[M][M]0. NONE[M]":
          & I(SR$,0;KI,"29",BC%,1,KI = 0 OR KI = 1 OR KI = 2):
          ON BC% >0 OR KI = 0 GOTO 20:
          N1 = 1:
          N2 = MN * (KI = 1) +  ME * (KI = 2):
          K$ = K$(KI):
          MX = N2

     344  POKE 34,3:
          HOME :
          PRINT "DEFINE"K$"[M][M]INSTRUCTIONS:[M]
             - ENTER VALUES NEEDED[M]
             - USE '<' TO DECREMENT INDEX[M]
             - USE '>' TO END ENTRY[M]":
          POKE 34,10:
          N$ = N$(KI):
          BW% = BW% * (KI = 1)

     346  HOME :
          FOR N = N1 TO N2:
               PRINT "[M]FOR"N$" #"N:
               ON KI GOSUB 320,322:
               N = N - (BC% >0) - (BC% = 1)
                  * (N >1) +  N2 * (BC% = 2):
          NEXT N:
          & I("[M][M]EDIT" +  K$ +  " (Y/N)";"N";A$,"30",BC%,1):
          ON A$ = "N" OR BC% >0 GOTO 348:
          & I("[M]START AT" +  N$ +  " NUMBER";N1;N1,"31",
             BC%,0,N1 < = MX):
          ON N1 >0 GOTO 346


     348  & I("[M]SAVE HOW MANY" +  N$;MX;N,"32",
             BC%,0,N > = 0 AND N < = MX):
          ON N = 0 OR BC% >0 GOTO 20:
          NN = NN * (KI = 2) +  N * (KI = 1):
          NE = NE * (KI = 1) +  N * (KI = 2):
          ON KI GOSUB 126,120:
          GOTO 342
```

(350-352) Adjust nodal position or reverse diagonal.

```
     350  RN = 0:
          GOSUB 136:
          ON BC% >0 GOTO 20:
          GOSUB 188:
          POKE 34,3

     352  & D(T):
          HOME :
          PRINT "[M]1. ADJUST NODE POSITIONS[M]
             2. REVERSE ELEMENT DIAGONALS[M][M]
             0. NONE[M]":
          & I(SR$,0;KI,"33",BC%,1,KI = 0 OR KI = 1 OR KI = 2):
          ON BC% >0 OR KI = 0 GOTO 20:
          BW% = BW% * (KI = 1):
          ON KI GOSUB 330,334:
          & D(T):
          ON KI GOTO 126,120

65535 REM [M][M]1FEB85[M]JRC/DCD

END-OF-LISTING
```

PREPROCESS.HEAT

(1,212-236) Cold start initialization.

```
1  TEXT :
   & L(255):
   P% = 10:
   S$ = "STARTING":
   E$ = "ENDING":
   P$ = " NODE ":
   U$ = "( LEFT ARROW  RIGHT ARROW S Q )  ND/EL  PRESS
      <S>AT ":
   SR$ = "... SELECT BY NUMBER <RETURN>":
   WR$ = "... WHEN READY, PRESS <RETURN>":
   D$ =  CHR$ (4):
   FI$ = "FILEINFO.TXT":
   GOTO 212:
   REM PRE(H)
```

(2,238-244) Warm restart.

```
2  GOTO 238
```

Wait for user response, clear screen, and continue.

```
4  PRINT :
   & B(20,20):
   VTAB 24:
   & I(WR$;"";A$,"1",BC%):
   HOME :
   RETURN
```

Free run pause.

```
6  ON Z < >2 GOTO 4:
   & B(10,10):
   & B(0,0,10):
   HOME :
   RETURN
```

Draw symbol at a node and get instruction.

```
8  NO =  NOT IN * BN%(N *  NOT IN,1) +  N * IN:
   VTAB 23:
   HTAB 36:
   PRINT "N"NO;:
   CALL  - 868:
   XDRAW 67 AT  FN PX(XN(NO)), FN PY(YN(NO)):
   GET A$:
   A =  ASC (A$):
   XDRAW 67 AT  FN PX(XN(NO)), FN PY(YN(NO)):
   N = N +   NOT IN *  FN M(B) +  IN *  FN O(NN):
   ON A$ < >"S" AND A$ < >"Q" GOTO 8:
   RETURN
```

Find node for boundary condition deletion.

```
10 ON NI = 0 GOTO 16:
   POKE 34,21:
   HOME :
   FOR NA = 1 TO NI:
      NO =  FN L(N):
      A0 = (BI%(NA,0) =  FN E(N) AND BI%(NA,2) = BD
            AND BI%(NA,1) = BT):
      ON A0 GOSUB 12:
   NEXT NA:
```

```
        HOME :
        RETURN
```

Delete a boundary condition at a node and update array.

```
    12  GOSUB 42:
        FOR I = NA TO NI - 1:
            FOR J = 0 TO 2:
                BI%(I,J) = BI%(I + 1,J):
            NEXT J:
            FOR J = 0 TO 1:
                BV(I,J) = BV(I + 1,J):
            NEXT J:
        NEXT I:
        FOR J = 0 TO 2:
            BI%(NI,J) = 0:
        NEXT J:
        FOR J = 0 TO 1:
            BV(NI,J) = 0:
        NEXT J:
        NI = NI - 1:
        RETURN
```

Add a boundary condition and plot its symbol.

```
    14  NI = NI + 1:
        NO = FN L(N):
        BI%(NI,0) = FN E(N):
        BI%(NI,1) = BT:
        BI%(NI,2) = BD:
        BV(NI,0) = (BT = 4) * HC:
        BV(NI,1) = VB:
        NA = NI:
        GOTO 42

    16  RETURN
```

Select a node for addition/deletion of a single boundary condition.

```
    18  HOME :
        N = (I1 * (N <I1 OR N >I2)
           + N * (N >= I1 AND N < = I2)) * NOT IN + IN
           * (N * (N >0 AND N < = NN)
           + (N <1 OR N >NN)):
        PRINT U$" DESIRED POINT":
        GOSUB 8:
        ON A$ = "Q" GOTO 16:
        ON AC GOSUB 14,10:
        GOTO 16
```

(20-22) Select a range of boundary nodes for addition or removal of the same boundary condition.

```
    20  HOME :
        N = I1 * (N <I1 OR N >I2)
           + N * (N > = I1 AND N < = I2):
        PRINT U$;:
        INVERSE :
        PRINT S$;:
        NORMAL :
        PRINT P$:
        GOSUB 8:
        ON A$ = "Q" GOTO 16:
        N1 = N:
        ON AC * (BT <3) GOSUB 14,10:
        HOME :
        PRINT U$;:
```

```
     INVERSE :
     PRINT E$;:
     NORMAL :
     PRINT P$:
     GOSUB 8:
     ON A$ = "Q" OR (N = N1) * (BT <3) GOTO 16:
     N2 = N

22   IF BT <3 THEN
          ON AC GOSUB 14,10:
        A = 21:
        N = N1:
        N = N1 +   FN M(B):
        ON N = N2 GOTO 16:
        FOR K = 1 TO NB:
            ON AC GOSUB 14,10:
            N = N +   FN M(B):
            K = K +  NB * (N = N2):
     NEXT K:
     GOTO 16
```

Select element for addition or removal of a boundary condition.

```
24   ON N1 = N2 GOTO 16:
     N = N1:
     A = 21:
     FOR K = 1 TO NB:
         ON AC GOSUB 14,10:
         N = N +   FN M(B):
         K = K +  NB * (N = N2):
     NEXT K:
     GOTO 16
```

Identify element at its centroid.

```
26   VTAB 23:
     HTAB 36:
     PRINT "E"EL;:
     CALL  - 868:
     X1 =  FN CX(EL):
     Y1 =  FN CY(EL):
     XDRAW 67 AT X1,Y1:
     GET A$:
     A =  ASC (A$):
     XDRAW 67 AT X1,Y1:
     N1 = N:
     N = EL:
     EL = EL +   FN O(NE):
     N = N1:
     ON A$ = "S" OR A$ = "Q" GOTO 16:
     GOTO 26
```

Addition or removal of boundary condition for elements.

```
28   HOME :
     EL = EL * (EL >0 AND EL < = NE)
        +  (EL <1 OR EL >NE):
     PRINT U$" DESIRED ELEMENT":
     GOSUB 26:
     ON A$ = "Q" GOTO 16:
     ON AC GOSUB 14,10:
     GOTO 16Qf
```

(30-38 Convert user-specified boundary conditions to equivalent nodal boundary conditions plus convection term.

```
30   N = BI%(NA,0):
     BD = BI%(NA,2):
```

```
    ON BD = 4 GOTO 36:
    A = 21:
    N0 = N + FN M(B):
    N1 = BN%(N,1):
    N2 = BN%(N0,1):
    X1 = XN(N1):
    Y1 = YN(N1):
    X2 = XN(N2):
    Y2 = YN(N2):
    L = FN LE(I):
    NX = (Y2 - Y1) / L:
    NY = (X1 - X2) / L:
    EL = BN%(N,2):
    A1 = L * ((BT = 4) - (BT = 3)) * (TH(EL)
       * (TE = 1) / 2 + P2 * (TE = 2) / 6)

32  FO = A1 * BV(NA,1) * (NX * (BD = 1)
       + NY * (BD = 2) + (BD = 3)):
    BC(N1,2) = BC(N1,2) + FO * ((TE = 1)
       + (2 * X1 + X2) * (TE = 2)):
    BC(N2,2) = BC(N2,2) + FO * ((TE = 1)
       + (X1 + 2 * X2) * (TE = 2)):
    BC(N1,1) = 1:
    BC(N2,1) = 1:
    ON BT <4 GOTO 16:
    H = H + 1:
    J = NN + H:
    BC(J,2) = BV(NA,0):
    BC(J,1) = EL:
    BC(J,0) = FN SI(N1):
    RETURN
```

Convert a nodal heat source or nodal temperature boundary condition to equivalent.

```
34  N = BI%(NA,0):
    BT = BI%(NA,1):
    BC(N,2) = BC(N,2) * (BT = 1) + BV(NA,1):
    BC(N,1) = BT:
    RETURN
```

(36-38) Convert a flux or convection boundary condition on an element face to equivalent nodal sources.

```
36  EL = BI%(NA,0):
    N1 = ND%(EL,1):
    N2 = ND%(EL,2):
    N3 = ND%(EL,3):
    A1 = XN(N2) * YN(N3) + XN(N3) * YN(N1) + XN(N1)
       * YN(N2) - XN(N2) * YN(N1) - XN(N3)
       * YN(N2) - XN(N1) * YN(N3):
    FOR K = 1 TO DE:
        N1 = ND%(EL,K):
        BC(N1,2) = BC(N1,2) + A1 * BV(NA,1) / 6:
        BC(N1,1) = 1:
    NEXT K

38  ON BT <4 GOTO 16:
    F = F + 1:
    K = NN + H + F:
    BC(K,2) = BV(NA,0):
    BC(K,1) = EL:
    BC(0,1) = F:
    RETURN
```

Optionally display the list of files saved.

```
40  & I("[M]CATALOG?";"N";A$,"2",0,1):
```

```
ON A$ = "N" GOTO 16:
PRINT D$"CATALOG"DR$(DR):
RETURN
```

(42-48) Draw a boundary condition symbol.

```
42  VB = BV(NA,1):
    BT = BI%(NA,1):
    BD = BI%(NA,2):
    ON BD = 4 AND (BT = 1 OR BT = 4) GOTO 48:
    NO = (BT >2) * BN%(BI%(NA,0)
       * (BT >2),1) +  BI%(NA,0) * (BT <3):
    ON BT <3 GOTO 46:
    A = 21:
    N0 = BN%(N +   FN M(B),1)
```

Draw a convection boundary condition symbol along an element side.

```
44  X1 = XN(NO):
    Y1 = YN(NO):
    X2 = XN(N0):
    Y2 = YN(N0):
    L =  FN LE(I):
    NX = (Y2 - Y1) / L:
    NY = (X1 - X2) / L:
    NU = (BD = 1) +  NX * (BD = 3) +  (BD = 4):
    NV = (BD = 2) +  NY * (BD = 3):
    ON BT = 3 GOTO 56:
    ROT= 16 * ( ABS (NY) > ABS (NX)):
    XDRAW 69 AT  FN PX((X1 +  X2) / 2) +  5
       * NU, FN PY((Y1 +  Y2) / 2) - 5 * NV:
    ROT= 0:
    RETURN
```

Draw a nodal heat source or nodal temperature boundary condition symbol.

```
46  SCALE= 1:
    FOR K = 0 TO 3:
        ROT= 16 * K:
        XDRAW (70 +  (VB <0)) * (BT = 1) +  72
           * (BT = 2) AT  FN PX(XN(NO)), FN PY(YN(NO)):
        K = K +  4 * (BT = 2):
    NEXT K:
    ROT= 0:
    RETURN
```

Draw a heat source or convection boundary condition symbol on the face of an element.

```
48  SY = (70 +  (VB <0)) * (BT <4)
       +  69 * (BT = 4):
    FOR K = 0 TO 3:
        ROT= 16 * K:
        XDRAW SY AT  FN CX(EL), FN CY(EL):
        K = K +  4 * (BT = 4):
    NEXT K:
    ROT= 0:
    RETURN
```

(50-52) Plot boundaries and label the positive directions of the axes.

```
50  B0 = B:
    B1 = I1:
    B2 = I2:
    HGR :
    & D(G):
    HCOLOR= 3:
```

```
    A = 21:
    FOR B = 1 TO NS:
        I1 = IS(2 * B - 1):
        I2 = IS(2 * B):
        FOR I = I1 TO I2:
            N = I:
            N1 = N +  FN M(B):
            HPLOT  FN PX(XN(BN%(N,1))), FN PY(YN(BN%(N,1)))
               TO  FN PX(XN(BN%(N1,1))), FN PY(YN(BN%(N1,1))):
    NEXT I,B:
    B = B0:
    I1 = B1:
    I2 = B2

52  SCALE= 1:
    ROT= 48:
    DRAW 70 AT 5,90:
    ROT= 0:
    DRAW 70 AT 130,155:
    DRAW  ASC (DI$(1)) - 31 AT 148,158:
    DRAW  ASC (DI$(2)) - 31 AT 2,75:
    RETURN
```

Draw an element of the grid and label or shade it.

```
54  FOR I = 1 TO DE:
        N1 = ND%(EL,I):
        N2 = ND%(EL, FN P2(I)):
        X1 =  FN PX(XN(N1)):
        Y1 =  FN PY(YN(N1)):
        X2 =  FN PX(XN(N2)):
        Y2 =  FN PY(YN(N2)):
        HPLOT X1,Y1 TO X2,Y2:
    NEXT I:
    L$ =  STR$ (EL):
    XP =  FN CX(EL):
    YP =  FN CY(EL):
    ON IO = 2 GOTO 62:
    & F(CO%(RE),OP%(RE),XP,YP):
    RETURN
```

Draw a surface flux boundary condition.

```
56  X1 = (BD <4) *  FN PX((X1 + X2) / 2)
      + (BD = 4) * ( FN CX(EL * (BD = 4)) - 4 * NU):
    Y1 = (BD <4) *  FN PY((Y1 + Y2) / 2)
      + (BD = 4) * ( FN CY(EL * (BD = 4)) + 6 * NV)
```

Draw convection boundary condition.

```
58  ROT= 0:
    FOR I = 1 TO 4:
        A0 = (I = 1 AND VB <0)
           + (I = 4 AND VB > = 0):
        SY = 64 * A0 + 74 *  NOT A0:
        SCALE= 1 + A0:
        XDRAW SY AT X1 + 2 * I * NU,Y1 - 3 * I * NV:
    NEXT I:
    SCALE= 1:
    RETURN
```

Draw all boundary condition symbols.

```
60  SCALE= 1:
    FOR NA = 1 TO NI:
        N = BI%(NA,0) * (BI%(NA,2) <4):
        EL = BI%(NA,0) * (BI%(NA,2) = 4):
```

```
      GOSUB 42:
   NEXT NA:
   RETURN
```

Draw L$ label on graphics screen.

```
62 FOR J1 = 1 TO  LEN (L$):
      XDRAW  ASC ( MID$ (L$,J1,1)) - 31 AT XP - 7
         *  LEN (L$) / 2 +  7 * (J1 - 1),YP +  3.5:
   NEXT J1:
   RETURN
```

Erase background for label L$.

```
64 HCOLOR= 0:
   FOR J1 = 1 TO 9:
      HPLOT XP - 3.5 *  LEN (L$) - 1,YP - 4.5
         +  J1 TO XP +  3.5 *  LEN (L$),YP - 4.5 +  J1:
   NEXT J1:
   HCOLOR= 3:
   GOTO 62
```

(66-96) Specify properties and indicate if property is uniform.

```
66 HOME :
   & M(MP = (1)):
   FOR I = 1 TO NE:
      MP(I,0) = I:
   NEXT I:
   & I("ISOTROPIC? (Y/N)[M]  (DOES " + TY$(1)
      + "[M]  EQUAL " + TY$(2)
      + "[M]          FOR EACH ELEMENT?)","Y";A$,"3",
      0,1):
   KE = (A$ < >"N"):
   T1 = T1 +  (KE = 1 AND T1 = 1):
   MP(0,3) = (TE = 2) +  MP(0,3) * (TE = 1)

68 FOR TY = T1 TO T2:
      E1 = 1:
      E2 = NE:
      R1 = 1:
      R2 = NQ:
      POKE 34,2:
      HOME :
      PRINT  TAB( (40 -  LEN (TY$(TY))
         - 10) / 2);"PROPERTY: ";:
      INVERSE :
      PRINT TY$(TY):
      NORMAL :
      POKE 34,3

70    PRINT "[M]"TY$(TY)" IS UNIFORM WITHIN[M][M]
         1. ENTIRE BODY[M]
         2. "IO$(1)"[M]
         3. "IO$(2)"[M]":
      & I(SR$,1;IO,"4",BC%,1,IO >0 AND IO <4):
      TY = TY * (BC% < >1) +  (BC% = 1)
         * (TY - 2) * (TY >T1):
      IF BC% = 1 THEN
            NEXT TY

72    ON BC% GOTO 66,92,70:
      UN = (IO = 1):
      ON UN < >1 GOTO 74:
      HOME :
      INVERSE :
      PRINT "[M]"TY$(TY)" IS UNIFORM.":
```

```
        NORMAL :
        & I("[M]WHAT IS THE " +  TY$(TY)
          +  "[M]VALUE";MP(1,TY);MP(1,TY),"5",BC%):
        ON BC% GOTO 70,92,70

74      E0 = 1:
        R0 = 1:
        MP(0,TY) = UN:
        IF UN = 1 THEN
            J1 = 1 +  (TY = 2 AND KE = 1):
            FOR EL = 1 TO NE:
                MP(EL,TY) = MP(1,TY):
                MP(EL,1) = MP(EL,J1):
            NEXT EL:
            FOR RE = 1 TO NQ:
                RP(RE,TY) = MP(1,TY):
                RP(RE,1) = RP(RE,J1):
            NEXT RE:
            GOTO 88
```

Input coded and non-coded values for a property.

```
76      HOME :
        IO = IO - 1:
        INVERSE :
        PRINT "[M]DEFINING "TY$(TY)"[M]
           UNIFORM BY "IO$(IO)"-":
        NORMAL :
        & I("[M]HOW MANY CODED " +  TY$(TY)
           +  "[M]VALUES ARE NEEDED (0-6)";2
           * (IO < >1 OR (IO = 1 AND R2 >6)) +  R2
           * (IO = 1 AND R2 <7);NC,"6",BC%,1,NC >
           = 0 AND NC <7):
        ON BC% GOTO 70,92,76

78      ON NC = 0 GOTO 80:
        PRINT "[M]INPUT A VALUE FOR EACH CODE:":
        INVERSE :
        PRINT " CODE      "TY$(TY):
        NORMAL :
        FOR I = 1 TO NC:
            & I("   " +   CHR$ (64 +  I) +  "     ";
               CV(I,TY);CV(I,TY),"7",BC%):
            I = I +  NC * (BC% = 2) - (1 +  (I >1))
               * (BC% = 1) - (BC% = 3):
        NEXT I:
        ON BC% = 2 GOTO 92

80      GOSUB 50:
        POKE 34,20:
        HOME :
        ON IO GOTO 100:
        ON NC >0 GOSUB 98:
        FOR EL = E1 TO E2:
            EO = EL:
            GOSUB 54:
            MP$ = "A":
            ON NC >0 GOTO 82:
            MP$ =  STR$ (MP(EL,TY))
```

Input properties for each element.

```
82          POKE 34,22:
            HOME :
            PRINT TY$(TY)" FOR ELEMENT "EL:
```

```
            & I("CODE/ # / Q ";MP$;A$,"8",BC%):
            ON BC% GOTO 82,86,82:
            ON LEN (A$) = 0 GOTO 86:
            A = ASC (A$):
            BC% = 2 * (A = 81):
            ON (A >47 AND A <58) OR
              A = 46 OR (A >64 AND A <65
              + NC) OR A = 81 GOTO 84:
            & B(200,20):
            GOTO 82

84          EL = (EL - 1) * (BC% = 1) + NE * (BC% = 2)
              + EL * (BC% = 0):
            ON BC% GOTO 82,86,82:
            MP(EL,TY) = VAL (A$)
              * (A <58 AND A >45)
              + CV((A - 64) * (A >64 AND A <65
              + NC),TY) * (A 64 AND A <65 + NC):
            IF TY = 2 AND KE = 1 THEN
               MP(EL,1) = MP(EL,2)

86          ON BC% = 0 GOSUB 54:
            L$ = CHR$ (A):
            ON A >64 AND BC% < >2 GOSUB 62:
            L$ = "*":
            ON A <65 AND BC% = 0 GOSUB 62:
            EL = EL + E2 * (BC% = 2):
        NEXT EL:
        GOSUB 4:
        & D(T):
        POKE 34,2:
        HOME
```

Edit scope of properties (by region or by element).

```
88      & I("[M]EDIT " + TY$(TY)
          + " VALUES";"N";A$,"9",0,1):
        ON A$ = "N" GOTO 92:
        PRINT "[M]EDIT SCOPE:[M][M]
          1. UNIFORMITY WITHIN "IO$(1)"[M]
          2. UNIFORMITY WITHIN "IO$(2)"[M]":
        & I(SR$,IO;IO,"10",BC%,1,IO >0 AND IO <3):
        ON BC% GOTO 88,92,88

90      EL = E0 * (E0 < = NE) + (E0 >NE):
        RE = R0 * (R0 < = NQ) + (R0 >NQ):
        A = EL * (IO = 2) + RE * (IO = 1):
        A1 = NE * (IO = 2) + NQ * (IO = 1):
        & I("[M]START AT WHAT "
          + IO$(IO);A;A,"11",BC%,0,A >
          = 0 AND A < = A1):
        RE = A:
        EL = A:
        ON A = 0 GOTO 74:
        E1 = EL:
        R1 = RE:
        IO = IO + 1:
        POKE 34,2:
        GOTO 76
```

Check uniformity.

```
92      POKE 34,2:
        HOME :
        PRINT "CHECKING UNIFORMITY...":
        & O(MP,1 TO NE,TY,A):
```

```
        MP(0,TY) = (MP(1,TY) = MP(NE,TY)):
        MP(0,1) = MP(0,1 +  (TY = 2 AND KE = 1)):
        & O(MP,1 TO NE,0,A):
        PRINT "[M][M]"TY$(TY)" IS"UN$(MP(0,TY))" UNIFORM":
        GOSUB 6:
        ON BC% GOTO 88:
        TY = TY * (BC% < >2) +  T2 * (BC% = 2):
    NEXT TY
```

(94-96) Select property to edit.

```
94  ON Z = 2 GOTO 16:
    HOME :
    T1 = (TY - 1) * (TY >0) +  (TY = 0)
       * (1 +  (KE = 1)):
    PRINT "[M][M]EDIT PROPERTIES:[M]":
    J1 = 2 +  (TE = 1):
    FOR I = 1 TO J1:
        PRINT I". "TY$(I):
    NEXT I:
    PRINT "[M]0. NONE[M]":
    & I(SR$,0;T1,"12",BC%,1,
       T1 > = 0 AND T1 <1 +  J1):
    ED = 1:
    T2 = T1:
    ON T1 +  1 GOTO 96,66,66,68

96  POKE 34,0:
    POKE 2166,2:
    PRINT "[M][M]* END OF MATERIAL PROPERTIES *":
    GOSUB 6:
    ON BC% >0 GOTO 94:
    RETURN
```

Display coded material properties.

```
98  POKE 34,20:
    HOME :
    FOR I = 1 TO NC:
        VTAB 21 +  (I >3):
        HTAB ((I - 3) * (I >3)
           +  I * (I <4)) * 12 - 10:
        PRINT  CHR$ (64 +  I)"="CV(I,TY):
    NEXT I:
    RETURN
```

(100-106) Specify properties by input region.

```
100 ON NC >0 GOSUB 98:
    FOR RE = R1 TO R2:
        R0 = RE:
        FOR EL = 1 TO NE:
            ON ND%(EL,0) = RE GOSUB 54:
        NEXT EL:
        XP = CR(RE,1):
        YP = CR(RE,2):
        L$ =  STR$ (RE):
        GOSUB 64:
        RP$ = "A":
        ON NC >0 GOTO 102:
        RP$ =  STR$ (RP(RE,TY))

102     POKE 34,22:
        HOME :
        PRINT TY$(TY)" FOR REGION "RE:
        & I("CODE / # / Q ";RP$;A$,"13",BC%):
        ON BC% GOTO 102,106,102:
```

```
          ON  LEN (A$) = 0 GOTO 106:
          A =  ASC (A$):
          BC% = 2 * (A = 81):
          ON (A >47 AND A <58) OR
             A = 46 OR (A >64 AND A <65
             +  NC) OR A = 81 GOTO 104:
          & B(200,20):
          GOTO 102

104       RE = (RE - 1) * (BC% = 1) +  NQ * (BC% = 2)
             +  RE * (BC% = 0):
          ON BC% GOTO 102,106,102:
          IF  LEN (A$) >0 THEN
              RP(RE,TY) =  VAL (A$) * (A <65 OR A >64
                 +  NC) +  CV((A - 64) * (A >64 AND A < = 64
                 +  NC),TY) * (A >64
                 AND A < = 64 +  NC):
              IF TY = 2 AND KE = 1 THEN
                  RP(RE,1) = RP(RE,2)

106       L$ =  STR$ (RE):
          A0 = (BC% < >2 AND A < >81):
          ON A0 GOSUB 62:
          L$ =  CHR$ (A * (A >64) +  42 * (A <65)):
          ON A0 GOSUB 62:
          RE = RE +  R2 * (A0 = 0):
      NEXT RE:
      J1 = 1 +  (TY = 2 AND KE = 1):
      FOR I = 1 TO NE:
          MP(I,TY) = RP(ND%(I,0),TY):
          MP(I,1) = MP(I,J1):
      NEXT I:
      GOSUB 4:
      & D(T):
      POKE 34,2:
      HOME :
      GOTO 88
```

(108-112) Initialize, give instructions, and plot boundary.

```
108  HOME :
     ON ED GOTO 110:
     NI = 0

110  POKE 34,2:
     & D(T):
     PRINT "INSTRUCTIONS (3.21):[M][M]
        BOUNDARY CONDITIONS  MAY BE  ENTERED  ORDELETED
        AT SINGLE NODES  OR OVER A RANGE OF LOCATIONS
        ON THE BOUNDARY.[M][M]FIRST, DEFINE THE B.C.
        TYPE  AND  VALUE,THEN SPECIFY THE RANGE OF
        APPLICATION.[M]"

112  PRINT "RANGES ARE DEFINED BY  COUNTER-CLOCKWISE
        (AS VIEWED FROM  INSIDE THE BODY)  MOVE-MENT
        FROM THE STARTING POINT TO THE END-ING POINT.[M][M]":
     GOSUB 6:
     ON BC% GOTO 16,16,110:
     GOSUB 50:
     B = 1:
     I1 = IS(2 * B - 1):
     I2 = IS(2 * B):
     ON ED GOSUB 60
```

(114-132) Define boundary type, direction, value, and range of application.

```
114  POKE 34,20:
     HOME :
     PRINT "DEFINE BOUNDARY CONDITIONS-[M]OPTIONS:
        ENTER, DELETE, QUIT, BRANCH":
     & I(" ( E D Q <>)";"E";A$,"14",BC%,1,
        A$ = "E" OR A$ = "D" OR A$ = "Q"):
     ON BC% GOTO 110,134,114:
     ON A$ = "Q" GOTO 134:
     AC = (A$ = "E") +  2 * (A$ = "D"):
     N = 1

116  POKE 34,20:
     HOME :
     PRINT AC$(AC)"-[M]1. "BT$(2) TAB( 20)"
        3. "BT$(4)"[M]2. "BT$(3) TAB( 20)"4. "BT$(1):
     POKE 34,21:
     & I(" ( 1 2 3 4 Q <>)";"1";A$,"15",BC%,1,
        A$ = "1" OR A$ = "2" OR A$ = "3" OR
        A$ = "4" OR A$ = "Q"):
     ON BC% GOTO 114,114,116

118  BT =  VAL (A$):
     BT = (BT = 4) +  (BT +  1) * (BT < >4):
     ON A$ = "Q" GOTO 114:
     BD = 0:
     ON BT GOTO 124,124,122:
     BD = 3:
     ON TE = 2 GOTO 124:
     POKE 34,20:
     HOME :
     PRINT AC$(AC)" "BT$(BT)"-":
     POKE 34,21:
     PRINT "1. "BD$(3) TAB( 20)"2. "BD$(4)

120  & I("( 1 2 Q <>)";"1";A$,"16",BC%,1,
        A$ = "1" OR A$ = "2" OR A$ = "Q"):
     ON BC% GOTO 116,114:
     ON A$ = "Q" GOTO 114:
     BD =  VAL (A$) +  2:
     GOTO 124

122  POKE 34,20:
     HOME :
     PRINT AC$(AC)" "BT$(BT)"-[M]1. "BD$(1) TAB( 20)"
        3. "BD$(3)"[M]2. "BD$(2) TAB( 20)FA$:
     & I(" (1 2 3" +  F$ +  " Q <>)";"3";
        A$,"17",BC%,1,( VAL (A$) >= 1
        AND  VAL (A$) < = 3 +  (TE = 1)) OR A$ = "Q"):
     ON BC% GOTO 116,114,122:
     ON A$ = "Q" GOTO 114:
     BD =  VAL (A$)

124  POKE 34,20:
     HOME :
     PRINT AC$(AC)" "BD$(BD)" "BT$(BT)"-":
     VTAB 22:
     POKE 34,21:
     IN = 0:
     IF BT <3 THEN
         & I(".. AT INTERIOR NODE(S)";"N";A$,"18",BC%,1):
        ON BC% GOTO 116,114,124:
        ON A$ = "Q" GOTO 114:
        IN = (A$ = "Y"):
        HOME
```

```
126  IF AC = 1 AND BT = 4 THEN
        HOME :
       & I("SURFACE CONDUCTANCE = ";HC;HC,"19",
          BC%,0,HC > = 0):
       ON BC% GOTO 116,114,126:
       & I("FLUID TEMP. = ";TF;TF,"20",BC%):
       ON BC% GOTO 126,114,126:
       POKE 34,20:
       HOME :
       PRINT BD$(BD)" "BT$(BT)": HC="HC" TF="TF:
       VB = HC * TF:
       POKE 34,21:
       GOTO 130

128  POKE 34,21:
     ON AC = 2 GOTO 130:
     HOME :
     & I("  " +  BT$(BT) +  " VALUE = ";0;VB,"21",BC%):
     ON BC% GOTO 116,114,128:
     POKE 34,20:
     HOME :
     PRINT BD$(BD)" "BT$(BT)" = "VB:
     POKE 34,21

130  ON IN GOSUB 18:
     ON IN GOTO 114:
     IF BT >2 THEN
        HOME :
       ON BD = 4 GOSUB 28:
       ON BD <4 GOSUB 20:
       GOTO 114

132  HOME :
     PRINT "SPECIFY WHERE TO "AC$(AC)" B.C.[M]
        1. SINGLE NODE,  OR  2.  CONTINUOUS":
     & I(" ( 1 2 Q <>)";"1";A$,"22",BC%,1,A$
        = "1" OR A$ = "2" OR A$ = "Q"):
     ON BC% GOTO 124,114,132:
     ON A$ = "Q" GOTO 114:
     A0 =  VAL (A$):
     ON OA GOSUB 18,20:
     GOTO 114
```

Specify boundary conditions on another boundary?

```
134  IF NS >1 THEN
        POKE 34,20:
       HOME :
       & I("ANOTHER BOUNDARY";"Y";A$,"23",BC%,1):
       HOME :
       IF A$ = "Y" THEN
          PRINT " BOUNDARY #";:
         GET A$:
         B =  VAL (A$):
         ON B <1 OR B >NS GOTO 134:
         I1 = IS(2 * B - 1):
         I2 = IS(2 * B):
         GOTO 114
```

(136-150) Edit boundary conditions, define composite source and node temperature equivalent boundary conditions.

```
136  & D(T):
     POKE 34,0:
     HOME :
```

```
     ON Z = 2 GOTO 138:
     PRINT "** EDIT BOUNDARY CONDITIONS **[M][M]":
     & I("CHANGES? (Y/N)";"N";A$,"24",0,1):
     ED = (A$ = "Y"):
     ON ED GOTO 108
```

Save boundary condition specification codes and boundary values (RE=10).

```
138  ON NI = 0 GOTO 16:
     & C(BI):
     DIM BI(NI,4):
     FOR I = 1 TO NI:
         FOR J = 0 TO 2:
             BI(I,J) = BI%(I,J):
         NEXT J:
         BI(I,3) = BV(I,0):
         BI(I,4) = BV(I,1):
     NEXT I:
     RE = 10:
     GOSUB 184:
     GOSUB 178:
     N1 = NI:
     N2 = 4:
     ONERR  GOTO 202

140  EF = 0:
     GOSUB 180:
     ON  PEEK (2163) = 3 GOTO 142:
     & S(BI,N$)
```

(142-146) Define equivalent nodal boundary conditions.

```
142  ON EF GOTO 140:
     GOSUB 182:
     & C(BI):
     SI%(10) = 0:
     GOSUB 6:
     PRINT "- BOUNDARY CONDITIONS MUST BE CONVERTED
        INTO EQUIVALENT SOURCES AND TEMPERATURESAT NODES
        (PLUS CONVECTION).[M]"

144  PRINT "[M]DEFINING NODAL B.C...";:
     H = 0:
     F = 0:
     FOR I = 1 TO NI:
         H = H +  (BI%(I,1) = 4 AND BI%(I,2) = 3):
         F = F +  (BI%(I,1) = 4 AND BI%(I,2) = 4):
     NEXT I:
     NP = NN +  H +  F:
     & C(BC):
     DIM BC(NP,2):
     FOR NA = 1 TO NI:
         BT = BI%(NA,1):
         PRINT ".";:
         ON BT GOSUB 34:
     NEXT NA
```

(146-150) Save equivalent nodal boundary conditions (RE = 11)

```
146  H = 0:
     F = 0:
     FOR NA = 1 TO NI:
         PRINT ".";:
         BT = BI%(NA,1):
         ON BT = 3 OR BT = 4 GOSUB 30:
     NEXT NA:
```

```
     BC(0,0) = H:
     FOR NA = 1 TO NI:
         PRINT ".";:
         BT = BI%(NA,1):
         ON BT = 2 GOSUB 34:
     NEXT NA:
     PRINT :
     RE = 11:
     GOSUB 184:
     GOSUB 178:
     N1 = NP:
     N2 = 2:
     ONERR  GOTO 202

148  EF = 0:
     GOSUB 180:
     ON  PEEK (2163) = 3 GOTO 150:
     & S(BC,N$)

150  ON EF GOTO 148:
     GOSUB 182:
     R1 = 12:
     R2 = 22:
     GOSUB 200:
     & C(BC,TH):
     SI%(11) = 0:
     RETURN
```

(152-158) Define plotting scale, find region centroids and normals to boundaries.

```
152  A = (SB - ST) / (VH - VL):
     SC = (SR - SL) / (UH - UL):
     SC = SC * (SC < = A) +  A * (SC >A)

154  NS = BN%(0,0):
     & C(CR):
     DIM CR(NQ,2):
     PRINT "[M]FINDING REGION CENTROIDS";:
     FOR EL = 1 TO NE:
         RE = ND%(EL,0):
         PRINT ".";:
         FOR N = 1 TO DE:
             N1 = ND%(EL,N):
             CR(RE,1) = CR(RE,1) +  XN(N1):
             CR(RE,2) = CR(RE,2) + YN(N1):
             CR(RE,0) = CR(RE,0) +  1:
     NEXT N,EL

156  FOR RE = 1 TO NQ:
         CR(RE,1) =  FN PX(CR(RE,1) / CR(RE,0)):
         CR(RE,2) =  FN PY(CR(RE,2) / CR(RE,0)):
     NEXT RE:
     FOR I = 1 TO 2 * NS:
         IS(I) = BN%(I,0):
     NEXT I:
     & C(NV):
     DIM NV(NB,2):
     PRINT "[M][M]DEFINING NORMALS TO BOUNDARIES";

158  N = 0:
     FOR B = 1 TO NS:
         I1 = IS(2 * B - 1):
         I2 = IS(2 * B):
```

```
    FOR I = I1 TO I2:
        PRINT ".";:
        N = N + 1:
        A = 8:
        N1 = N + FN M(B):
        A = 21:
        N2 = N + FN M(B):
        B1 = BN%(N1,1):
        B2 = BN%(N2,1):
        X1 = XN(B1):
        Y1 = YN(B1):
        X2 = XN(B2):
        Y2 = YN(B2):
        L = FN LE(I):
        NV(N,1) = (Y2 - Y1) / L:
        NV(N,2) = (X1 - X2) / L:
    NEXT I,B:
    PRINT :
    RETURN
```

(160-172) Retrieve data files for element nodes (RE=5), nodal coordinates (RE=6), unique line nodes (RE=7), and boundary nodes (RE=8).

```
160 RE = 5:
    GOSUB 184:
    GOSUB 178:
    NE = N1:
    DE = N2:
    & C(ND%):
    DIM ND%(NE,DE):
    ON SI%(RE) = 0 GOTO 174:
    ONERR GOTO 202

162 EF = 0:
    GOSUB 188:
    ON BC% GOTO 16,16,160:
    & R(ND%,N$):
    ON EF GOTO 162:
    & O(ND%,1 TO N1,0,A):
    NQ = ND%(N1,0):
    & R(ND%,N$):
    GOSUB 196:
    RE = 6:
    GOSUB 178:
    NN = N1:
    & C(XY,XN,YN):
    DIM XY(NN,2),XN(NN),YN(NN):
    ON SI%(RE) = 0 GOTO 174

164 EF = 0:
    GOSUB 188:
    ON BC% GOTO 160,16,160:
    & R(XY,N$):
    ON EF GOTO 164:
    GOSUB 196

166 FOR I = 1 TO NN:
        XN(I) = XY(I,1):
        YN(I) = XY(I,2):
    NEXT I:
    & O(XY,1 TO NN,1,A):
    XL = XY(1,1):
    XH = XY(NN,1):
    & O(XY,1 TO NN,2,A):
```

```
         YL = XY(1,2):
         YH = XY(NN,2):
         & C(XY)

168  RE = 7:
     GOSUB 178:
     NL = N1:
     & C(LN%):
     DIM LN%(NL,3):
     ON SI%(RE) = 0 GOTO 174

170  EF = 0:
     GOSUB 188:
     ON BC% GOTO 160,16,168:
     & R(LN%,N$):
     ON EF GOTO 170:
     GOSUB 196:
     RE = 8:
     GOSUB 178:
     NB = N1:
     & C(BN%):
     DIM BN%(NB,2):
     ON SI%(RE) = 0 GOTO 174

172  EF = 0:
     GOSUB 188:
     ON BC% GOTO 168,16,168:
     & R(BN%,N$):
     ON EF GOTO 172:
     GOTO 196
```

(174-176) Error messages.

```
174  PRINT :
     FLASH :
     ON PEEK (222) = 77 GOTO 176:
     PRINT "[G][G]DATA NOT DEFINED":
     NORMAL :
     PRINT "EITHER FILE":
     INVERSE :
     PRINT NA$:
     NORMAL :
     PRINT "DOESN'T EXIST OR THE DATA IS NOT VALID":
     GOSUB 4:
     BC% = 2:
     RETURN

176  PRINT "[G][G][M]SORRY, OUT OF MEMORY":
     NORMAL :
     END
```

Read file descriptors.

```
178  PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"READ"FI$",R"RE:
     INPUT SI%(RE),NA$,DE$,N1,D1$,N2,D2$:
     PRINT D$"CLOSE"FI$:
     N$ = NA$ + DR$(DR):
     RETURN
```

Get data filename.

```
180  ON Z = 2 OR PEEK (2163) = 3 GOTO 190:
     PRINT "[M]*** ";:
     INVERSE :
     PRINT "SAVING";:
```

```
     NORMAL :
     PRINT " DATA TO DISKETTE ***":
     ON Z GOTO 194:
     PRINT "[M][M]FILE NAME FOR[M]"DE$:
     & I("";NA$,NA$,"25",BC%,30):
     ON BC% < >0 GOTO 180:
     GOSUB 192:
     N$ = NA$ +  DR$(DR):
     RETURN
```

Set enable flag, update file information and notify user.

```
182  SI%(RE) = 1:
     ON  PEEK (2163) = 3 GOTO 16:
     PRINT D$;"OPEN"FI$",L100"DR$(DR):
     PRINT D$"WRITE"FI$",R"RE:
     PRINT SI%(RE):
     PRINT NA$:
     PRINT DE$:
     PRINT N1:
     PRINT D1$:
     PRINT N2:
     PRINT D2$:
     PRINT D$"CLOSE"FI$:
     PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2" "D2$"[M]
        WERE SAVED":
     RETURN
```

(184-186) Verify existence of data file.

```
184  ONERR  GOTO 210

186  E3 = 0:
     PRINT D$:
     ON E3 GOTO 186:
     RETURN
```

(188-196) Request name of data file to retrieve.

```
188  ON Z = 2 GOTO 190:
     PRINT "[M]*** ";:
     INVERSE :
     PRINT "LOADING";:
     NORMAL :
     PRINT " DATA FROM DISKETTE ***[M][M]":
     ON Z GOTO 194:
     & I("FILE NAME FOR[M]" +  DE$,NA$,NA$,"26",BC%,30):
     ON BC% < >0 GOTO 188:
     GOSUB 192

190  N$ = NA$ +  DR$(DR):
     RETURN

192  ON  LEFT$ (NA$, LEN (KW$)) = KW$ GOTO 16:
     NA$ = KW$ +  "/" +  NA$:
     RETURN

194  N$ = NA$ +  DR$(DR):
     PRINT " ";NA$:
     RETURN
```

Print descriptors

```
196  ON Z = 2 GOTO 16:
     PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2" "D2$"[M]
        WERE LOADED":
     RETURN
```

Read and print problem description.

```
198  PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"READ"FI$",R0":
     INPUT TE,KW$,PD$:
     PRINT D$"CLOSE"FI$:
     PRINT "[M][M]PROBLEM KEYWORD: "KW$"[M][M]
        PROBLEM DESCRIPTION:[M]"PD$"[M][M]
        ELEMENT TYPE:[M] "TE$(TE)T$"[M][M]":
     RETURN
```

Disable data files.

```
200  ON  PEEK (2163) = 3 GOTO 16:
     PRINT D$"OPEN"FI$",L100"DR$(DR):
     FOR RE = R1 TO R2:
         PRINT D$"WRITE"FI$",R"RE:
         PRINT 0:
         SI%(RE) = 0:
     NEXT RE:
     PRINT D$"CLOSE"FI$:
     RETURN
```

(200-210) Error traps.

```
202  EF = 1:
     POKE 216,0:
     ER =  PEEK (222):
     ON ER = 255 GOTO 206:
     ON ER = 77 GOTO 176:
     PRINT :
     FLASH :
     PRINT  CHR$ (7) CHR$ (7)"ERROR # ";ER:
     NORMAL :
     PRINT :
     ER = ER * (ER >3 AND ER <11)
        +   NOT (ER >3 AND ER <11):
     ON ER GOTO 204:
     ON ER = 5 GOTO 204:
     ON ER >3 AND ER <11 GOTO 208

204  GOSUB 178:
     RESUME

206  END

208  PRINT ER$(ER - 4):
     ON ER = 6 GOSUB 40:
     ON ER = 4 OR ER = 5 OR ER = 8 OR ER = 9 GOSUB 4:
     ON ER <10 GOTO 204:
     & I("UNLOCK? (Y/N)";"Y";A$,"27",0,1):
     ON A$ = "N" GOTO 204:
     PRINT D$;"UNLOCK ";NA$;DR$(DR):
     GOTO 204

210  E3 = 1:
     POKE 216,0:
     ON  PEEK (222) = 255 GOTO 206:
     PRINT  CHR$ (7) CHR$ (7)"[M]
        DATA DISKETTE IS NOT IN DRIVE":
     GOSUB 4:
     RESUME
```

(1,212-236) Cold start initialization and title screen.

```
212  & D(T):
     POKE 2164,3:
```

```
     HOME :
     INVERSE :
     FOR I = 1 TO 4:
        PRINT  TAB( 2)" " TAB( 38)" ":
     NEXT I:
     NORMAL :
     VTAB 2:
     HTAB 3:
     PRINT " APPLIED FINITE ELEMENT ANALYSIS: ":
     HTAB 3:
     PRINT "     AN APPLE " CHR$ (221) CHR$ (219)
       " IMPLEMENTATION      ":
     VTAB 6:
     HTAB 12:
     PRINT "COPYRIGHT, 1985"

214  VTAB 8:
     HTAB 11:
     INVERSE :
     PRINT "PREPROCESS";:
     NORMAL :
     PRINT " (3.00)[M][M]ABSTRACT:[M][M]
        DEFINE MATERIAL PROPERTIES AND BOUNDARY
        CONDITIONS FOR THE PROBLEM.  MATERIAL
        PROPERTIES MUST BE DEFINED  BEFORE B. C."

216  PRINT "[M]MATERIAL PROPERTIES INCLUDE
        THERMAL      CONDUCTIVITIES AND ELEMENT THICKNESS[M]
        (FOR 2-DIMENSIONAL PROBLEMS).
        BOUNDARYCONDITION  OPTIONS  ARE:
        HEAT  SOURCES,SPECIFIED  TEMPERATURES,  HEAT  FLUXES,
        AND  CONVECTION.":
     GOSUB 6

218  & C(ER$,DR$):
     DIM ER$(6),DR$(2):
     ER$(0) = "DISKETTE IS WRITE PROTECTED":
     ER$(2) = "FILE NOT FOUND":
     ER$(3) = "VOLUME MISMATCH":
     ER$(4) = "I/O ERROR":
     ER$(5) = "DISK IS FULL":
     ER$(6) = "FILE IS LOCKED":
     DR$(1) = ",D1":
     DR$(2) = ",D2":
     DR =  PEEK (2048):
     ON DR = 2 GOTO 222

220  & B(200,20):
     PRINT "INSERT ";:
     FLASH :
     PRINT "DATA";:
     NORMAL :
     PRINT " DISKETTE INTO DRIVE":
     GOSUB 4

222  Z =  PEEK (2163):
     & C(TE$):
     DIM TE$(2):
     TE$(1) = "2-DIMENSIONAL":
     TE$(2) = "AXISYMMETRIC":
     T$ = " 3-NODE TRIANGLES":
     GOSUB 184:
     HOME :
     GOSUB 198:
```

```
    DEF  FN L(N) = ( NOT IN * BN%(N *  NOT IN,1) +  N * IN)
       * (BD <4) +  EL * (BD = 4):
    DEF  FN E(N) = (N * (BD <4) +  EL * (BD = 4))
       * (BT 2) +  NO * (BT <3)

224 DEF  FN P2(I) = I +  1 - DE * (I = DE):
    DEF  FN M(B) = ((N <I2) - (I2 - I1) * (N = I2))
       * (A = 21) - ((N I1) - (I2 - I1) * (N = I1)) * (A = 8):
    DEF  FN O(NA) = (A = 21) * (1 - NA * (N = NA)) - (A = 8)
       * (1 - NA * (N = 1)):
    P2 = 8 *  ATN (1):
    DEF  FN SI(N) = (ND%(EL,1) = N) +  2 * (ND%(EL,2) = N) +  3
       * (ND%(EL,3) = N):
    DIM SI%(22)

226 DEF  FN CX(EL) =  FN PX((XN(ND%(EL,1)) +  XN(ND%(EL,2))
       +  XN(ND%(EL,3))) / 3):
    DEF  FN CY(EL) =  FN PY((YN(ND%(EL,1)) +  YN(ND%(EL,2))
       +  YN(ND%(EL,3))) / 3)

228 DIM TY$(3),CV(6,3),CO%(10),OP%(10),NE$(1):
    TF$ = KW$ +  "/" +  "TEMPFILE":
    & M(CV = (1)):
    NE$(0) = "ND"

230 NE$(1) = "EL":
    DIM UN$(1),SF$(1),BT$(4),DF$(4),DI$(2),
       AC$(2),BD$(4):
    UN$(0) = " NOT":
    SF$(0) = "S":
    SF$(1) = "F":
    DATA 0,0,1,0,1,1,8,0,8,1,16,1,5,1,19,0,10,1,20,0:
    FOR K = 1 TO 10:
       READ CO%(K),OP%(K):
    NEXT K:
    GOSUB 160:
    DB =  INT (1.6 * NN):
    ON BC% >0 GOTO 212

232 ST = 20:
    SB = 140:
    SL = 30:
    SR = 250:
    DEF  FN LE(I) =  SQR ((X1 - X2) ^ 2 +  (Y1 - Y2) ^ 2):
    DEF  FN PX(X) = SL +  SC * (X - UL):
    DEF  FN PY(Y) = SB - SC * (Y - VL):
    UL = XL:
    UH = XH:
    VL = YL:
    VH = YH:
    GOSUB 152:
    DIM IO$(2):
    IO$(1) = "INPUT REGION":
    IO$(2) = "ELEMENT"

234 DF$(1) = "X":
    DF$(2) = "Y":
    DF$(3) = "R":
    DF$(4) = "Z":
    DI$(1) = DF$(2 * TE - 1):
    DI$(2) = DF$(2 * TE):
    AC$(1) = "ENTER":
    AC$(2) = "DELETE":
    BD$(1) = DI$(1) +  "-DIRECTION":
```

```
       BD$(2) = DI$(2) +  "-DIRECTION":
       BD$(3) = "BOUNDARY (OUT)":
       BD$(4) = "FACE (OUT)"

236    A$ = "THERMAL CONDUCTIVITY":
       TY$(1) = DI$(1) +  "-" +  A$:
       TY$(2) = DI$(2) +  "-" +  A$:
       TY$(3) = "THICKNESS":
       BT$(0) = "NO EXTERNAL FLOW":
       BT$(1) = "NODE HT SOURCE":
       BT$(2) = "NODE TEMP":
       BT$(3) = "SURFACE FLUX":
       BT$(4) = "CONVECTION":
       IF TE = 1 THEN
          F$ = " 4":
          FA$ = "4. " +  BD$(4)
```

(2,238-246) Warm restart; properties and boundary conditions menu.

```
238    NM =  PEEK (2166):
       TEXT :
       & D(T):
       HOME :
       EO = 1:
       ON Z = 2 AND  PEEK (2166) = 3 GOTO 246

240    PRINT "*** PROPERTIES & BOUND. COND. (3.05) ***[M]
         --- DATA INPUT ---[M]
         1. INPUT MATERIAL THERMAL PROPERTIES AND THICKNESS[M]
         2. INPUT TEMPERATURE, FLUX, CONVECTION
         &   SOURCE BOUNDARY CONDITIONS[M]"

242    PRINT " --- DATA EXAMINATION ---[M]
         3. LIST PROPERTIES[M]
         4. PLOT PROPERTIES[M]
         5. LIST INPUT BOUNDARY CONDITIONS[M]
         6. LIST EQUIV. BOUNDARY CONDITIONS[M]
         7. PLOT INPUT BOUNDARY CONDITIONS[M]
         8. PLOT EQUIV. BOUNDARY CONDITIONS[M][M]
         0. NONE OF THE ABOVE[M]"

244    & I(SR$,NM;MO,"28",BC%,1,MO >
         = 0 AND MO <9):
       ON BC% GOTO 212,246,238:
       ON MO = 0 GOTO 246:
       ON MO GOSUB 256,270,284,292,308,314,320,324:
       GOTO 238
```

(246-254) Exit menu.

```
246    HOME :
       HTAB 5:
       PRINT "*** PREPROCESS: EXIT (3.9) ***[M][M]
         1. PROCEED TO 'SOLVE' [M]
         2. EXIT TO MAIN MENU[M]
         3. EXIT TO 'CREATE TEXT FILE'[M]
         4. DON'T EXIT; REMAIN IN 'PREPROCESS'[M][M]
         0. STOP[M]"

248    & I(SR$,1;EO,"29",BC%,1,EO > = 0 AND EO <5):
       ON EO = 0 OR BC% = 2 GOTO 206:
       ON EO = 4 GOTO 238

250    POKE 2166,1:
       & C(PN$):
       DIM PN$(3):
```

```
      PN$(1) = "SOLVE":
      PN$(2) = "HELLO":
      PN$(3) = "CREATE TEXT FILE"

252  IF DR = 1 OR E0 = 3 THEN
        & B(200,20):
       PRINT "[M]INSERT ";:
       FLASH :
       PRINT "PROGRAM";:
       NORMAL :
       PRINT " DISKETTE CONTAINING[M]'";:
       INVERSE :
       PRINT PN$(EO);:
       NORMAL :
       PRINT "' INTO DRIVE":
       GOSUB 4:
       ON BC% >0 GOTO 246
```

For 'solve' relinquish graphics capability for data space.

```
254  POKE 103,63 * (EO = 1) +    PEEK (103)
       * (EO <>1):
     POKE 104,25 * (EO = 1) +    PEEK (104)
       * (EO < >1):
     POKE 6462,0 * (EO = 1) +    PEEK (6462)
        * (EO < >1):
     PRINT D$"RUN "PN$(EO)",D1"
```

(256-268) Option 1: input thermal properties and thicknesses.

```
256  POKE 34,0:
     HOME :
     PRINT "***     ENTRY OF PROPERTIES (3.1)      ***":
     POKE 34,2:
     HOME :
     & C(MP,RP):
     DIM MP(NE,3),RP(NQ,3):
     & I("RETRIEVE PROPERTIES FROM DISK";"N";
       A$,"30",BC%,1):
     ON BC% GOTO 16,16,256:
     ED = (A$ = "Y"):
     ON ED = 0 GOTO 262
```

(258-260) Retrieve material thermal properties (RE=9)

```
258  RE = 9:
     GOSUB 184:
     GOSUB 178:
     ON SI%(RE) = 0 GOTO 174:
     & C(MP):
     DIM MP(N1,N2):
     ONERR  GOTO 202

260  EF = 0:
     GOSUB 188:
     & R(MP,N$):
     ON EF GOTO 260:
     GOSUB 196:
     GOSUB 6:
     ON MO = 3 GOTO 286:
     ON MO = 4 GOTO 294
```

(262-266) Save material properties (RE=9) and thickness (temporary file).

```
262  T1 = 1:
     T2 = 4 - TE:
     FOR I = 1 TO NE:
        MP(I,0) = I:
```

```
     NEXT I:
     ON ED +  1 GOSUB 66,94:
     RE = 9:
     GOSUB 184:
     GOSUB 178:
     N1 = NE:
     N2 = 3:
     ONERR  GOTO 202

264  EF = 0:
     GOSUB 180:
     ON  PEEK (2163) = 3 GOTO 268:
     & S(MP,N$)

266  ON EF GOTO 264:
     GOSUB 182:
     R1 = 12:
     R2 = 22:
     GOSUB 200:
     PRINT D$"OPEN"TF$DR$(DR):
     PRINT D$"DELETE"TF$:
     PRINT D$"OPEN"TF$:
     PRINT D$"WRITE"TF$:
     FOR I = 0 TO NE:
         PRINT MP(I,3):
     NEXT I:
     PRINT D$"CLOSE"TF$

268  & C(MP,RP):
     SI%(9) = 0:
     POKE 2166,2:
     & I("[M][M]PROCEED TO BOUNDARY CONDITIONS","Y";
       A$,"31",0,1):
     ON A$ = "N" GOTO 16
```

(270-282) Menu option 2: input temperature, flux, convection, and source conditions.

```
270  POKE 34,0:
     HOME :
     PRINT "***  ENTRY OF BOUNDARY COND. (3.2)    ***[M]":
     POKE 34,2

272  GOSUB 184:
     RE = 9:
     GOSUB 178:
     IF SI%(RE) = 0 THEN
         PRINT  CHR$ (7) CHR$ (7)"[M]
            MATERIAL PROPERTIES MUST BE DEFINED     BEFORE
            BOUNDARY CONDITIONS.":
         GOSUB 4:
         GOTO 256

274  & C(TH,BI%,BV):
     DIM TH(NE),BI%(DB,2),BV(DB,1):
     PRINT D$"OPEN"TF$;DR$(DR):
     PRINT D$"READ"TF$:
     FOR I = 0 TO NE:
         INPUT TH(I):
     NEXT I:
     PRINT D$"CLOSE"TF$

276  HOME :
     & I("RETRIEVE BOUNDARY CONDITIONS FROM DISK","N";
       A$,"32",BC%,1):
```

```
       ON BC% GOTO 16,16,256:
       ED = (A$ = "Y"):
       ON ED = 0 GOTO 282
```

(278-280) Retrieve boundary conditions.

```
278    RE = 10:
       GOSUB 178:
       ON SI%(RE) = 0 GOTO 174:
       NI = N1:
       & C(BI):
       DIM BI(NI,4):
       ONERR  GOTO 202

280    EF = 0:
       GOSUB 188:
       & R(BI,N$):
       ON EF GOTO 280:
       GOSUB 196:
       GOSUB 6:
       ON MO = 5 GOTO 310:
       ON MO = 7 GOTO 322:
       FOR I = 1 TO NI:
           FOR J = 0 TO 2:
               BI%(I,J) = BI(I,J):
           NEXT J:
           BV(I,0) = BI(I,3):
           BV(I,1) = BI(I,4):
       NEXT I:
       & C(BI):
       SI%(10) = 0

282    GOSUB 108:
       POKE 2166,3:
       POKE 2165,4:
       & C(BI%,BV):
       GOTO 6
```

(284-290) Menu option 3: list properties; load first if necessary.

```
284    POKE 34,0:
       HOME :
       PRINT "*******  LIST PROPERTIES (3.3)   *******";
       POKE 34,2:
       T2 = 3 * (TE = 1) +  2 * (TE = 2):
       GOTO 258

286    HOME :
       & L(P%):
       FOR TY = 1 TO T2:
           POKE 34,2:
           HOME :
           PRINT  TAB( (40 -  LEN (TY$(TY))
              - 10) / 2)"PROPERTY: ";
           INVERSE :
           PRINT TY$(TY)"[M]":
           IF MP(0,TY) < >1 THEN
               PRINT "ELEMENT  "TY$(TY):
               POKE 34,5:
               HOME :
               VTAB 6:
               NORMAL :
               FOR EL = 1 TO NE:
                   PRINT "    "EL TAB( 12)MP(EL,TY):
                   NEXT EL
```

```
288      IF MP(0,TY) = 1 THEN
             PRINT TY$(TY)" IS UNIFORM.":
            NORMAL :
            PRINT "[M]FOR ALL ELEMENTS (1 TO "NE") THE[M]
               "TY$(TY)" VALUE IS ";:
            INVERSE :
            PRINT MP(1,TY):
            NORMAL

290       GOSUB 4:
         TY = TY - (BC% = 1) +  3 * (BC% = 2):
     NEXT TY:
     & L(255):
     POKE 2166,4:
     & C(MP):
     RETURN
```

(292-306) Menu option 4: Plot properties, retrieving data if necessary.

```
292  POKE 34,0:
     HOME :
     PRINT "******     PLOT PROPERTIES (3.4)     ******":
     POKE 34,2:
     T2 = 3 * (TE = 1) +  2 * (TE = 2):
     T3 = T2 +  1:
     GOTO 258

294  HOME :
     PRINT "** SELECT PROPERTIES TO PLOT **":
     FOR I = 1 TO T2:
        PRINT I". "TY$(I):
     NEXT I:
     PRINT "[M]"T3". ALL OF THESE[M][M]0. NONE[M]":
     & I(SR$,0;TY,"33",BC%,1,TY > = 0 AND TY < = T3):
     ON BC% >0 OR TY = 0 GOTO 16:
     T1 = (TY = T3) +  TY * (TY < >T3):
     T2 = T2 * (TY = T3) +  TY * (TY < >T3)

296  FOR TY = T1 TO T2:
        GOSUB 298:
        TY = TY +  3 * (BC% = 2) - (2 * (TY >1)
          +  (TY = 1)) * (BC% = 1):
     NEXT TY:
     & C(MP,PV):
     POKE 2166,5:
     RETURN

298  POKE 34,20:
     HOME :
     PRINT  TAB( (40 -  LEN (TY$(TY))
        - 10) / 2)"PROPERTY: ";:
     INVERSE :
     PRINT TY$(TY):
     NORMAL :
     POKE 34,21:
     HOME :
     ON MP(0,TY) GOTO 306:
     HGR :
     & D(G):
     HCOLOR= 3:
     & C(PV):
     DIM PV(9)

300  FOR I = 1 TO NL:
        N1 = LN%(I,1):
```

```
            N2 = LN%(I,2):
            HPLOT  FN PX(XN(N1)), FN PY(YN(N1))
               TO  FN PX(XN(N2)), FN PY(YN(N2)):
        NEXT I:
        J = 1:
        NC = 0:
        & O(MP,1 TO NE,TY,A)

302     NC = NC +  1:
        PV(NC) = MP(J,TY):
        & W((MP,1 TO NE,TY),PV(NC),PV(NC),N1%,BR%,ER%):
        PRINT :
        FOR I = BR% TO ER%:
            EL = MP(I,0):
            & F(CO%(NC),OP%(NC), FN CX(EL), FN CY(EL)):
        NEXT I:
        ON ER% = NE GOTO 304:
        J = ER% +  1:
        GOTO 302

304     HOME :
        POKE 34,0:
        FOR I = 1 TO NC:
            VTAB 22 +  (I >3) +  (I >6):
            HTAB ((I - 6) * (I >6) +  (I - 3)
               * (I >3 AND I <7)
               +  I * (I <4)) * 12 - 10:
            PRINT  CHR$ (64 +  I)"="PV(I):
        NEXT I:
        PRINT "(A=DENSE TO " CHR$ (64 +  NC)"=SPARSE DOTS)";:
        & I("  <RETURN>";"";A$,"34"):
        RETURN

306     GOSUB 50:
        & F(1,1, FN CX(1), FN CY(1)):
        PRINT "[M]"TY$(TY)" IS UNIFORM.[M]
           FOR ALL ELEMENTS (1 TO "NE") THE[M]VALUE IS ";:
        INVERSE :
        PRINT MP(1,TY);:
        NORMAL :
        & I("  <RETURN>";"";A$,"35"):
        RETURN
```

(308-318) Menu option 5: List the boundary conditions.

```
308     POKE 34,0:
        HOME :
        PRINT "* LIST INPUT BOUNDARY CONDITIONS (3.5) *":
        POKE 34,2:
        GOTO 278

310     HOME :
        & L(P%):
        PRINT "INPUT B.C. SPECIFICATIONS:":
        INVERSE :
        PRINT "NODE/EL   B.C. DIRECTION & TYPE   VALUE":
        NORMAL :
        POKE 34,5:
        HOME

312     FOR I = 1 TO NI:
            N = BI(I,0):
            BT = BI(I,1):
            BD = BI(I,2):
            NO = N * (BT <3 OR BD = 4) +  BN%(N
```

```
            * (BT >2 AND BD <4),1)
            * (BT >2 AND BD <4):
         PRINT NE$(BD = 4)"#"NO; TAB( 7)BD$(BD)" "
            BT$(BT)" " TAB( 34)BI(I,4):
    NEXT I:
    & L(255):
    POKE 34,0:
    POKE 2166,6:
    & C(BI):
    SI%(10) = 0:
    GOTO 4
```

(314-318) Menu option 6: List equivalent boundary conditions.

```
314 POKE 34,0:
    HOME :
    PRINT "LIST EQUIVALENT BOUNDARY CONDITIONS(3.6)":
    POKE 34,2:
    GOTO 326

316 HOME :
    & L(P%):
    PRINT "EQUIV. B.C.:":
    INVERSE :
    PRINT "NODE      B.C. TYPE        B.C. VALUE[M]":
    NORMAL :
    POKE 34,5:
    HOME :
    VTAB 6:
    FOR I = 1 TO NN:
         PRINT " "I TAB( 8)BT$(BC(I,1)) TAB( 25)BC(I,2):
    NEXT I:
    POKE 34,2:
    IF H +  F = 0 THEN
          & L(255):
         POKE 34,0:
         POKE 2166,7:
         GOTO 332

318 GOSUB 4:
    PRINT "CONVECTION B. C.:":
    INVERSE :
    PRINT "ELEM-SIDE      CONVECT. COEFF.[M]":
    NORMAL :
    POKE 34,5:
    VTAB 6:
    FOR I = NN +  1 TO NP:
         I1 = (I >NN +  H):
         PRINT " "BC(I,1)"-"SF$(I1)BC(I,0) TAB( 17)BC(I,2):
    NEXT I:
    & L(255):
    POKE 34,0:
    POKE 2166,7:
    GOTO 332
```

(320-322) Menu option 7: Plot the user-specified boundary conditions.

```
320 POKE 34,0:
    HOME :
    PRINT "* PLOT INPUT BOUNDARY CONDITIONS (3.7) *":
    POKE 34,2:
    & C(BI%,BV):
    DIM BI%(DB,2),BV(DB,1):
    GOTO 278

322 HOME :
```

```
     VTAB 21:
     PRINT  TAB( 7)"INPUT BOUNDARY CONDITIONS":
     & C(BI%,BV):
     DIM BI%(DB,2),BV(DB,1):
     FOR I = 1 TO NI:
         FOR J = 0 TO 2:
             BI%(I,J) = BI(I,J):
         NEXT J:
         BV(I,0) = BI(I,3):
         BV(I,1) = BI(I,4):
     NEXT I:
     & C(BI):
     SI%(10) = 0:
     GOSUB 50:
     GOSUB 60:
     POKE 2166,8:
     & C(BI%,BV):
     GOTO 4
```

(324-336) Menu option 8: Plot the equivalent boundary conditions.

```
324  POKE 34,0:
     HOME :
     PRINT "PLOT EQUIVALENT BOUNDARY CONDITIONS(3,8)":
     POKE 34,2

326  RE = 11:
     GOSUB 184:
     GOSUB 178:
     NP = N1:
     & C(BC):
     DIM BC(NP,2):
     ON SI%(RE) = 0 GOTO 174:
     ONERR  GOTO 202

328  EF = 0:
     GOSUB 188:
     & R(BC,N$):
     ON EF GOTO 328:
     H = BC(0,0):
     F = BC(0,1):
     GOSUB 196

330  GOSUB 4:
     ON MO = 6 GOTO 316:
     HOME :
     VTAB 21:
     PRINT  TAB( 7)"EQUIV. BOUNDARY CONDITIONS":
     GOSUB 50:
     FOR NO = 1 TO NN:
         BT = BC(NO,1):
         VB = BC(NO,2):
         ON BT >0 GOSUB 46:
     NEXT NO:
     ON H +  F = 0 GOTO 332:
     BT = 4:
     FOR I = 1 TO H +  F:
         BD = 3 +  (I >H):
         ON BD - 2 GOSUB 334,336:
     NEXT I

332  & C(BC):
     SI%(11) = 0:
     GOTO 4
```

```
334  A = BC(NN +  I,0):
     EL = BC(NN +  I,1):
     NO = ND%(EL,A):
     N0 = ND%(EL, FN P2(A)):
     GOSUB 44:
     RETURN

336  EL = BC(NN +  I,1):
     ROT= 0:
     SCALE= 1:
     XDRAW 69 AT  FN CX(EL), FN CY(EL):
     RETURN

65535 REM [M][M]1FEB85[M]JRC/DCD

END-OF-LISTING
```

SOLVE.HEAT

(1,194-208) Cold start initialization.

```
1 CLEAR :
  TEXT :
  & D(T):
  P% = 10:
  & L(255):
  SR$ = "... SELECT BY NUMBER <RETURN>":
  WR$ = "WHEN READY, PRESS <RETURN>":
  D$ = CHR$ (4):
  FI$ = "FILEINFO.TXT":
  POKE 2164,4:
  Z = PEEK (2163):
  GOTO 194:
  REM SOLVE(H)
```

(2,210-218) Warm restart.

```
2 GOTO 210
```

Pause for user response, clear screen, and continue.

```
4 ON FR = 1 GOSUB 228:
  VTAB (FR = 1) + 24 * (FR = 0):
  CALL - 958:
  ON FR = 1 GOTO 16:
  & B(20,20):
  & I(WR$;"";A$,"1",BC%):
  HOME :
  RETURN
```

(6-10) Select level of display details.

```
6 ON MO < >2 GOTO 4:
  & B(20,20):
  & I("[M]<D>CHANGES DISPLAY OPTION,
     <RETURN>";"";A$,"2",BC%):
  HOME :
  ON LEFT$ (A$,1) < >"D" GOTO 16:
  POKE 34,2:
  HOME

8 PRINT "*** DISPLAY OPTIONS ***[M][M]
     1. ONLY FINAL GLOBAL MATRICES[M]
     2. ELEMENT & GLOBAL MATRICES[M]
     3. DETAIL DURING CALCULATION[M][M]
     0. NO DISPLAY":
  & I(SR$,1;PO,"3",BC%,1,PO > = 0 AND PO <4):
  POKE - 16368,0:
  POKE 34,2:
  HOME :
  IF PO >1 OR EL = 0 THEN
       POKE 35,24:
       HOME :
       RETURN

10 ON FR GOTO 16:
   VTAB 20:
   PRINT "  PRESS <ESC>TO CHANGE DISPLAY OPTION":
   RETURN
```

(12-16) Decompose global equations using Gauss elimination method.

```
12 FOR I = 1 TO NN - 1:
   VTAB 5:
```

```
            PRINT "  WORKING ON ROW #"I" OF "NN:
            G1 = GS(I,1):
            MJ = I +  BW - 1:
            MJ = MJ * (MJ < = NN) +  NN * (MJ >NN):
            N = 0:
            MK = BW:
            IF NN - I +  1 <BW THEN
                MK = NN - I +  1

14          FOR J = I +  1 TO MJ:
                GOSUB 32:
                MK = MK - 1:
                N = N +  1:
                G2 = GS(I,N +  1):
                FOR K = 1 TO MK:
                    GS(J,K) = GS(J,K)
                       - G2 * GS(I,N +  K) / G1:
                NEXT K:
                GF(J) = GF(J) - G2 * GF(I) / G1:
            NEXT J:
            PRINT :
        NEXT I

16  RETURN
```

(18-24) Modify global equations for constant temperature boundary condition.

```
18  FOR DF = 1 TO NN:
        VTAB 5:
        HTAB 3:
        PRINT "WORKING ON ROW "DF" OF "NN:
        ON BC(DF,1) < >2 GOTO 24:
        BV = BC(DF,2):
        K = DF - 1:
        FOR J = 2 TO BW:
            M = DF +  J - 1:
            IF M < = NN THEN
                GF(M) = GF(M) - GS(DF,J) * BV:
                GS(DF,J) = 0

20          IF K >0 THEN
                GF(K) = GF(K) - GS(K,J) * BV:
                GS(K,J) = 0:
                K = K - 1

22      NEXT J:
        GS(DF,1) = GS(DF,1) * (GS(DF,1) >
            = .0005) +  50000000 * (GS(DF,1) <.0005):
        GF(DF) = GS(DF,1) * BV
24  NEXT DF:
    RETURN
```

Define material property (conductivity) matrix for element EL.

```
26  & M(D = (0)):
    D(1,1) = MP(EL,1):
    D(2,2) = MP(EL,2):
    ON PO <3 GOTO 16:
    & L(0):
    PRINT "[M]MATERIAL PROPERTY (CONDUCTIVITY) MATRIX":
    FOR I = 1 TO DD:
        PRINT :
        INVERSE :
        PRINT " " TAB( 39)" [M]ROW "I:
        NORMAL :
        PRINT " "D(I,1) TAB( 18)D(I,2):
```

```
     NEXT I:
     & L(255):
     GOTO 6
```

Define shape function gradient matrix for element EL.

```
28   FOR K = 1 TO DE:
        ND(K) = EN%(EL,K):
        X(K) = NC(ND(K),1):
        Y(K) = NC(ND(K),2):
     NEXT K:
     RB =  SQR ((X(1) ^ 2 +  X(2) ^ 2 +  X(3) ^ 2 +  X(1)
        * X(2) +  X(1) * X(3) +  X(2) * X(3)) / 6)

30   A2 = 2 *  FN AR(EL):
     & M(B = (0)):
     FOR J = 1 TO DE:
        I =  FN P1(J):
        K =  FN P2(J):
        B(1,J) = (Y(K) - Y(I)) / A2:
        B(2,J) = (X(I) - X(K)) / A2:
     NEXT J:
     ON PO = 3 GOSUB 36:
     RETURN
```

Display moving '' and check for 'esc" key to abort calculation.*

```
32   POKE 35,24:
     INVERSE :
     VTAB 12 +  V:
     HTAB 19 +  H:
     PRINT " ":
     H1 = (H = V):
     H = H *  NOT H1 +   NOT H * H1:
     V = V * H1 +   NOT V *  NOT H1:
     VTAB 12 +  V:
     HTAB 19 +  H:
     PRINT "*":
     NORMAL :
     ON  PEEK ( - 16384) < = 127 GOTO 16:
     POKE 34,2:
     HOME :
     GOTO 8
```

List temperature and coordinates for mode I.

```
34   & L(P%):
     PRINT "[M] NODE "I" TEMPERATURE IS "UU(I):
     HTAB 10:
     PRINT "X OR R = "NC(I,1):
     HTAB 10:
     PRINT "Y OR Z = "NC(I,2):
     & L(255):
     RETURN
```

(36-38) List nodal coordinates, element area, shape function, and gradient matrix for element EL.

```
36   & L(0):
     PRINT "[M]ELEMENT NO. "EL"  NODE LOCATIONS":
     FOR I = 1 TO DE:
        PRINT " NODE "ND(I)":   "X(I) TAB( 25)Y(I):
     NEXT I:
     PRINT "[M]ELEMENT AREA = "A2 / 2"[M][M][M][M]
        SHAPE FUNCTION GRADIENT MATRIX":
     FOR I = 1 TO DD:
        INVERSE
```

```
38      PRINT " " TAB( 39)" [M]DNJ/DX"I:
        NORMAL :
        FOR J = 1 TO DK:
            PRINT  TAB( 13 * J - 12)B(I,J)CR$(J = DK);:
    NEXT J,I:
    & L(255):
    GOTO 6
```

(40-62) Menu option 2: Assemble global force (source) and stiffness (coefficient) matrices and save global matrices.

```
40  TEXT :
    HOME :
    PRINT "** FORCE & STIFFNESS MATRICES (4.2)    **[M][M]":
    POKE 34,2:
    & C(D,ND,X,Y,B,BT,KE,C,GF,GS):
    DIM D(DD,DD),ND(DE),X(DE),Y(DE),B(DD,DK),BT(DK,DD),
       KE(DK,DK),C(DK,DD),GF(NN),GS(NN,BW):
    PO = 0:
    EL = 0:
    ON FR = 0 GOSUB 8:
    H = 0:
    V = 0:
    EL = 1

42  ON UN GOSUB 26:
    FOR EL = 1 TO NE:
        POKE 34,2:
        POKE 35,24 - 14 * (PO <2 AND EL >1):
        HOME :
        PRINT "ASSEMBLING ELEMENT "EL" OF "NE" ELEMENTS...":
        IF PO <2 AND EL = 1 THEN
             GOSUB 10
```

Define element stiffness matrix.

```
44      ON UN = 0 GOSUB 26:
        GOSUB 28:
        & M(BT = TRN(B)):
        & M(C = BT.D):
        & M(KE = C.B):
        VE = (MP(EL,3) * (TE = 1) +  2 * PI * RB
           * (TE = 2)) * A2 / 2:
        & M(KE = KE * (VE)):
        ON PO <2 OR BC% = 2 GOTO 48:
        & L(0)

46      PRINT "[M]STIFFNESS MATRIX FOR ELEMENT "EL"[M]":
        INVERSE :
        FOR I = 1 TO DK:
            PRINT  TAB( 13 * I - 11)"COLUMN "ND(I);:
        NEXT I:
        PRINT  TAB( 39)" ":
        NORMAL :
        FOR I = 1 TO DK:
            PRINT "ROW #"ND(I):
            FOR K = 1 TO DK:
                PRINT  TAB( 13 * K - 12)" "KE(I,K);:
            NEXT K:
            PRINT :
        NEXT I:
        & L(255):
        ON PO = 2 GOSUB 6
```

Assemble element stiffness matrix into global stiffness matrix.

```
48      FOR I = 1 TO DK:
```

```
                ON PO <2 GOSUB 32:
                KI = ND(I):
                FOR J = 1 TO DK:
                    KJ = ND(J):
                    IF KJ > = KI THEN
                        K = KJ - KI + 1:
                        GS(KI,K) = GS(KI,K) + KE(I,J):
                        IF PO >2 THEN
                            & L(0):
                            PRINT "UPDATED K("KI","KJ")"
                              TAB( 19)"IS "GS(KI,K):
                            & L(255)

50          NEXT J,I:
            POKE 34,(2 * (PO >2) + PEEK (34)
               * (PO <3)):
            ON PO >2 GOSUB 6:
            ON PO >2 GOSUB 140:
            ON PO >2 GOSUB 142:
            IF PO >1 THEN
                 PRINT
```

Assemble nodal source into global soure matrix.

```
52  NEXT EL:
    FOR I = 1 TO NN:
        GF(I) = BC(I,2) * (BC(I,1) = 1):
    NEXT I:
    POKE 34,2:
    HOME :
    PRINT "****** GLOBAL MATRICES ASSEMBLED ******":
    GOSUB 4:
    IF PO >0 THEN
         GOSUB 140:
        GOSUB 142
```

(54-56) Save initial heat source vector (RE=12).

```
54  POKE 34,2:
    HOME :
    RE = 12:
    GOSUB 160:
    GOSUB 152:
    N1 = NN:
    N2 = 1:
    ONERR GOTO 180

56  EF = 0:
    GOSUB 154:
    ON PEEK (2163) = 3 GOTO 58:
    & S(GF,N$)
```

(58-60) Save initial global stiffness matrix (RE=13).

```
58  ON EF GOTO 56:
    GOSUB 156:
    RE = 13:
    GOSUB 152:
    N1 = NN:
    N2 = BW

60  EF = 0:
    GOSUB 154:
    ON PEEK (2163) = 3 GOTO 62:
    & S(GS,N$)
```

Disable invalid data files.

```
62  ON EF GOTO 60:
    GOSUB 156:
    R1 = 14:
    R2 = 22:
    GOSUB 178:
    POKE 34,0:
    POKE 2166,3:
    GOTO 4
```

(64-84) Menu option 3: Apply convection boundary conditions (modify global matrices as required by convection boundary condition).

```
64  POKE 34,0:
    HOME :
    PRINT "** APPLY CONVECTION BOUND COND (4.3)  **[M][M]":
    POKE 34,2:
    RE = 13:
    ON SI%(RE) GOTO 70:
    GOSUB 160:
    GOSUB 152:
    ON SI%(RE) GOTO 66:
    PRINT "[M][G][G]GLOBAL SOURCE AND STIFFNESS MATRICES[M]
      MUST BE DEFINED BEFORE USING CONVECTION B.C.":
    GOSUB 4:
    GOTO 40
```

(66-68) If necessary, retrieve the initial global stiffness matrix (RE=13).

```
66  ONERR  GOTO 180

68  EF = 0:
    GOSUB 164:
    & R(GS,N$):
    ON EF GOTO 68:
    GOSUB 172:
    NN = N1:
    BW = N2:
    GOSUB 4

70  HOME :
    PO = 0:
    ON FR GOTO 72:
    PRINT "**** DISPLAY OPTIONS (4.35)****[M][M]
      1. INITIAL GLOBAL STIFFNESS MATRIX[M]
      2. NEW GLOBAL STIFFNESS MATRIX[M]
      3. BOTH GLOBAL STIFFNESS MATRICES[M][M]
      0. NO DISPLAY[M]":
    & I(SR$,0;PO,"4",BC%,1,PO > = 0 AND PO <4)

72  LA$ = "":
    ML = 2:
    ON PO GOSUB 100:
    LA$ = "----- ORIGINAL MATRIX -----":
    ON PO = 3 GOSUB 100:
    NO = NN +  HB:
    ON NN = NP GOTO 78:
    ON HB = 0 GOTO 76:
    FOR I = NN +  1 TO NO:
        EL = BC(I,1):
        S = BC(I,0):
        N1 = EN%(EL,S):
        N2 = EN%(EL, FN P2(S)):
        X1 = NC(N1,1):
        X2 = NC(N2,1):
```

```
          L =  SQR ((X1 - X2) ^ 2 +  (NC(N1,2)
             - NC(N2,2)) ^ 2)

 74       HA = BC(I,2) * L * (MP(EL,3) * (TE = 1)
             +  PI * (TE = 2)) / 6:
          GS(N1,1) = GS(N1,1) +  HA * (2 * (TE = 1)
             +  (3 * X1 +  X2) * (TE = 2)):
          GS(N2,1) = GS(N2,1) +  HA * (2 * (TE = 1)
             +  (X1 +  3 * X2) * (TE = 2)):
          J = N1 * (N1 <N2) +  N2 * (N2 <N1):
          K =  ABS (N2 - N1) +  1:
          GS(J,K) = GS(J,K) +  HA * ((TE = 1)
             +  (X1 +  X2) * (TE = 2)):
      NEXT I

 76   ON HF = 0 GOTO 78:
      FOR I = NO +  1 TO NP:
          EL = BC(I,1):
          A = BC(I,2) *  FN AR(EL) / (4 * DE):
          FOR J = 1 TO DE:
              N1 = EN%(EL,J):
              N2 = EN%(EL, FN P2(J)):
              GS(N1,1) = GS(N1,1) +  A:
              GS(N2,1) = GS(N2,1) +  A:
              K = N1 * (N1 <N2) +  N2 * (N2 <N1):
              L =  ABS (N2 - N1) +  1:
              GS(K,L) = GS(K,L) +  A:
      NEXT J,I

 78   LA$ = "":
      ML = 2:
      ON PO = 2 GOSUB 100:
      LA$ = "       -------- NEW MATRIX --------":
      ON PO = 3 GOSUB 100
```

Save stiffness matrix modified for convection boundary conditions (RE=14).

```
 80   HOME :
      RE = 14:
      GOSUB 160:
      GOSUB 152:
      N1 = NN:
      N2 = BW:
      ONERR  GOTO 180

 82   EF = 0:
      GOSUB 154:
      ON  PEEK (2163) = 3 GOTO 84:
      & S(GS,N$)
```

Disable data files.

```
 84   ON EF GOTO 82:
      GOSUB 156:
      R1 = 16:
      R2 = 22:
      GOSUB 178:
      POKE 34,0:
      POKE 2166,4:
      GOTO 4
```

(86-112) Menu option 4: Apply constant temperature boundary conditions (load global matrices modified by convection, modify for constant temperature boundary conditions and save modified matrices).

```
 86   POKE 34,0:
```

```
    HOME :
    PRINT "** APPLY TEMP. BOUNDARY COND. (4.4) **[M][M][M]":
    POKE 34,2:
    RE = 14:
    ON SI%(RE) = 1 GOTO 92:
    GOSUB 160:
    GOSUB 152:
    ON SI%(RE) = 1 GOTO 88:
    PRINT  CHR$ (7) CHR$ (7)"[M]APPLY CONVECTION B.C.
       BEFORE APPLYING[M]TEMPERATURE B.C.":
    GOSUB 4:
    GOTO 64

88  ONERR  GOTO 180
```

Retrieve, if necessary, the stiffness matrix modified for convection boundary conditions (RE=14).

```
90  EF = 0:
    GOSUB 164:
    NN = N1:
    BW = N2:
    & R(GS,N$):
    ON EF GOTO 90:
    GOSUB 172

92  RE = 12:
    ON SI%(RE) = 1 GOTO 96:
    GOSUB 160:
    GOSUB 152:
    ON SI%(RE) = 0 GOTO 86:
    ONERR  GOTO 180
```

Retrieve, if necessary, the initial heat source vector (RE=12).

```
94  EF = 0:
    GOSUB 164:
    NN = N1:
    & R(GF,N$):
    ON EF GOTO 94:
    GOSUB 172:
    GOSUB 4
```

Select display detail.

```
96  HOME :
    PO = 0:
    ON FR GOTO 102:
    PRINT "**** DISPLAY OPTIONS (4.45) ****[M][M]
       1. ORIGINAL STIFFNESS MATRIX[M]
       2. MODIFIED STIFFNESS MATRIX[M]
       3. ORIGINAL SOURCE VECTOR[M]
       4. MODIFIED SOURCE VECTOR[M]
       5. ALL OF THE ABOVE (OPT 1-4)"

98  PRINT "6. MODIFIED STIFFNESS & SOURCE
       VECTOR        (OPT 2 & 4)[M][M]0. NO DISPLAY[M]":
    & I(SR$,0;PO,"5",BC%,1,PO > = 0 AND PO <7):
    ML = 2:
    LA$ = " ------ ORIGINAL MATRIX ------":
    ON PO = 1 OR PO = 5 GOSUB 100:
    HOME :
    ML = 1:
    ON PO = 3 OR PO = 5 GOSUB 100:
    HOME :
    GOTO 102
```

```
100  HOME :
     PRINT LA$:
     POKE 34,4:
     ON ML GOSUB 140,142:
     RETURN

102  GOSUB 18:
     LA$ = " -------- MODIFIED SOURCE VECTOR -------":
     ML = 1:
     ON PO = 4 OR PO = 5 OR PO = 6 GOSUB 100:
     IF PO = 2 OR PO = 5 OR PO = 6 THEN
         HOME :
        PRINT " ----- MODIFIED STIFFNESS MATRIX -----":
        POKE 34,4:
        GOSUB 142
```

(104-106)Save sources array modified by temperature boundary conditions (RE=15).

```
104  POKE 34,2:
     HOME :
     RE = 15:
     GOSUB 160:
     GOSUB 152:
     SI%(RE) = 1:
     N1 = NN:
     N2 = 1:
     ONERR  GOTO 180

106  EF = 0:
     GOSUB 154:
     ON  PEEK (2163) = 3 GOTO 108:
     & S(GF,N$)
```

(108-110)Save stiffness array modified by temperature boundary conditions (RE=16).

```
108  ON EF GOTO 106:
     GOSUB 156:
     RE = 16:
     GOSUB 152:
     SI%(RE) = 1:
     N1 = NN:
     N2 = BW

110  EF = 0:
     GOSUB 154:
     ON  PEEK (2163) = 3 GOTO 112:
     & S(GS,N$)
```

Disable subsequent data files.

```
112  ON EF GOTO 110:
     GOSUB 156:
     R1 = 17:
     R2 = 22:
     GOSUB 178:
     POKE 34,0:
     POKE 2166,5:
     GOTO 4
```

(114-132) Menu option 5: Solve for nodal temperatures (Retrieve, if necesary, the global matrices modified for temperature boundary conditions (RE=15,16) and decompose the global equation, solve for nodal temperatures, and save the results.

```
114  POKE 34,0:
     HOME :
     PRINT "* SOLVE FOR NODE TEMPERATURES (4.5) *[M][M][M]":
     POKE 34,2:
     RE = 15:
```

```
        ON SI%(RE) = 1 GOTO 124:
        GOSUB 160:
        GOSUB 152:
        ON SI%(RE) = 1 GOTO 118

116     PRINT :
        PRINT  CHR$ (7) CHR$ (7)"CONSTANT TEMPERATURE
           BOUNDARY CONDITIONSMUST BE APPLIED BEFORE SOLVING
           FOR NODE TEMPERATURES":
        GOSUB 4:
        GOTO 86

118     ONERR  GOTO 180

120     EF = 0:
        GOSUB 164:
        & R(GF,N$):
        ON EF GOTO 120:
        GOSUB 172:
        NN = N1:
        RE = 16:
        GOSUB 152:
        ON SI%(RE) = 0 GOTO 116

122     EF = 0:
        GOSUB 164:
        & R(GS,N$):
        ON EF GOTO 122:
        GOSUB 172:
        NN = N1:
        BW = N2:
        GOSUB 4

124     HOME :
        PRINT "[M]DECOMPOSING MATRICES...":
        H = 0:
        V = 0:
        GOSUB 12:
        HOME :
        PRINT "SOLVING FOR TEMPERATURES...[M]":
        POKE 34,3:
        PO = 0:
        & C(UU):
        DIM UU(NN):
        IF FR = 0 THEN
            & I("DISPLAY NODE TEMPERATURES AS OBTAINED",
               "N";A$,"6",BC%,1):
            ON BC% < >0 GOTO 16:
            PO = (A$ = "Y"):
            HOME

126     I = NN:
        UU(I) = GF(I) / GS(I,1):
        ON PO = 1 GOSUB 34:
        FOR K = 1 TO NN - 1:
            ON PO = 0 GOSUB 32:
            I = NN - K:
            MJ = BW * (I +  BW - 1 < = NN)
               +  (NN - I +  1) * (I +  BW - 1 >NN):
            SU = 0:
            FOR J = 2 TO MJ:
                N = I +  J - 1:
                SU = SU +  GS(I,J) * UU(N):
            NEXT J:
```

```
        UU(I) = (GF(I) - SU) / GS(I,1):
        ON PO = 1 GOSUB 34:
     NEXT K:
     POKE 34,2:
     ON PO GOSUB 4:
     HOME
```

(128-130) Save nodal temperatures (RE=17).

```
128  RE = 17:
     GOSUB 160:
     GOSUB 152:
     N1 = NN:
     N2 = 1:
     ONERR  GOTO 180

130  EF = 0:
     GOSUB 154:
     ON  PEEK (2163) = 3 GOTO 132:
     & S(UU,N$)

132  ON EF GOTO 130:
     GOSUB 156:
     R1 = 18:
     R2 = 22:
     GOSUB 178:
     POKE 34,0:
     POKE 2166,6:
     GOTO 4
```

(134-138) Menu option 6: List nodal temperatures.

```
134  FR = 0:
     POKE 34,0:
     HOME :
     PRINT "***     LIST TEMPERATURES (4.6)        ***[M][M]":
     POKE 34,2:
     RE = 17:
     ON SI%(RE) = 1 GOTO 138:
     GOSUB 160:
     GOSUB 152:
     & C(UU):
     DIM UU(N1):
     ON SI%(RE) = 0 GOTO 150:
     ON BC% = 2 GOTO 16:
     ONERR  GOTO 180
```

Retrieve, if necessary, the temperature array (RE=17).

```
136  EF = 0:
     GOSUB 164:
     & R(UU,N$):
     ON EF GOTO 136:
     NN = N1:
     GOSUB 172:
     GOSUB 4

138  HOME :
     & L(P%):
     PRINT "NODE  TEMPERATURE[M]"U$:
     POKE 34,4:
     FOR I = 1 TO NN:
        PRINT " "I TAB( 8)UU(I):
     NEXT I:
     & L(255):
     GOTO 4
```

(140-142) List the global source vector and global stiffness matrix.

```
140  HOME :
     & L(P%):
     INVERSE :
     PRINT "     GLOBAL FORCE (SOURCE) VECTOR       ":
     NORMAL :
     PRINT :
     POKE 34, PEEK (37):
     FOR I = 1 TO NN:
          PRINT " "GF(I)CR$(I = NN);:
     NEXT I:
     & L(255):
     GOSUB 6:
     POKE 34,2:
     RETURN
```

List the global stiffness matrix.

```
142  HOME :
     & L(P%):
     PRINT "GLOBAL STIFFNESS MATRIX[M]":
     POKE 34, PEEK (37):
     FOR I = 1 TO NN:
          INVERSE :
          PRINT "ROW "I; TAB( 39)" ":
          NORMAL :
          FOR J = 1 TO BW:
               PRINT GS(I,J)CR$(J = BW);:
     NEXT J,I:
     & L(255):
     GOSUB 6:
     POKE 34,2:
     RETURN
```

(144-148) Retrieve renumbered element node numbers (RE=5) and coordinates of the renumbered nodes.

```
144  RE = 5:
     GOSUB 160:
     GOSUB 152:
     ON SI%(RE) = 0 GOTO 150:
     ONERR  GOTO 180

146  EF = 0:
     GOSUB 164:
     & R(EN%,N$):
     ON EF GOTO 146:
     GOSUB 172:
     RE = 6:
     GOSUB 152:
     ON SI%(RE) = 0 GOTO 150

148  EF = 0:
     GOSUB 164:
     & R(NC,N$):
     ON EF GOTO 148:
     GOTO 172
```

Print error messages.

```
150  PRINT :
     FLASH :
     PRINT  CHR$ (7) CHR$ (7)"DATA NOT DEFINED":
     NORMAL :
     PRINT "EITHER FILE":
     INVERSE :
     PRINT NA$:
```

```
       NORMAL :
       PRINT "DOESN'T EXIST OR THE DATA IS NOT VALID":
       GOSUB 4:
       BC% = 2:
       RETURN
```

Read descriptors for record RE.

```
152  N1 =  FRE (0):
     PRINT D$:
     PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"READ"FI$",R"RE:
     INPUT SI%(RE),NA$,DE$,N1,D1$,N2,D2$:
     PRINT D$"CLOSE"FI$:
     GOTO 166
```

Get the filename for the data array to be saved.

```
154  ON Z = 2 OR  PEEK (2163) = 3 GOTO 166:
     PRINT "[M]*** ";:
     INVERSE :
     PRINT "SAVING";:
     NORMAL :
     PRINT " DATA TO DISKETTE ***":
     ON FR GOTO 166:
     ON Z GOTO 170:
     & I("[M]FILE NAME FOR[M]" +  DE$,NA$,NA$,"7",BC%,30):
     ON BC% < >0 GOTO 154:
     GOSUB 168:
     GOTO 166
```

Set active status for data and save information for record RE.

```
156  SI%(RE) = 1:
     ON  PEEK (2163) = 3 GOTO 16:
     PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"WRITE"FI$",R"RE:
     PRINT SI%(RE):
     PRINT NA$:
     PRINT DE$:
     PRINT N1:
     PRINT D1$:
     PRINT N2:
     PRINT D2$:
     PRINT D$"CLOSE"FI$
```

Print descriptors.

```
158  ON Z =  2 GOTO 16:
     PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2" "D2$"[M]
        WERE SAVED"CR$(FR = 0):
     ON FR = 0 GOTO 16:
     PRINT " IN FILE: "NA$:
     RETURN
```

(160-162) Verify existence of data file.

```
160  ONERR  GOTO 190

162  E2 = 0:
     PRINT D$:
     PRINT D$"CKFILE"FI$;DR$(DR):
     ON E2 GOTO 162:
     RETURN
```

(164-170) Request a filename for data being saved.

```
164  ON Z = 2 GOTO 166:
     PRINT "[M]*** ";:
```

```
     INVERSE :
     PRINT "LOADING";:
     NORMAL :
     PRINT " DATA FROM DISKETTE ***":
     ON FR GOTO 166:
     ON Z GOTO 170:
     & I("[M]FILE NAME FOR[M]" +  DE$,NA$,NA$,"8",BC%,30):
     ON BC% < >0 GOTO 164:
     GOSUB 168

166  N$ = NA$ +  DR$(DR):
     RETURN

168  ON  LEFT$ (NA$, LEN (KW$)) = KW$ GOTO 16:
     NA$ = KW$ +  "/" +  NA$:
     RETURN

170  PRINT " ";NA$:
     GOTO 166
```

Print loaded data descriptors.

```
172  ON Z = 2 GOTO 16:
     PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2" "D2$"[M]
        WERE LOADED"CR$(FR = 0):
     ON FR = 0 GOTO 16:
     PRINT " FROM FILE: "NA$:
     RETURN
```

Read information for record RE.

```
174  PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"READ"FI$",R0":
     INPUT TE,KW$,PD$:
     PRINT D$"CLOSE"FI$:
     RETURN
```

Print problem description.

```
176  ON  LEN (KW$) = 0 GOSUB 174:
     PRINT "[M]PROBLEM KEYWORD: "KW$"[M][M]
        PROBLEM DESCRIPTION:[M]"PD$:
     RETURN
```

Disable subsequent data files.

```
178  ON  PEEK (2163) = 3 GOTO 16:
     PRINT D$"OPEN"FI$",L100"DR$(DR):
     FOR RE = R1 TO R2:
         PRINT D$"WRITE"FI$",R"RE:
         PRINT 0:
         SI%(RE) = 0:
     NEXT RE:
     PRINT D$"CLOSE"FI$:
     RETURN
```

(180-192)Error handling.

```
180  EF = 1:
     POKE 216,0:
     ER =  PEEK (222):
     ON ER = 255 GOTO 184:
     FLASH :
     ON ER = 77 GOTO 186:
     PRINT  CHR$ (7) CHR$ (7)"[M]ERROR # ";ER;"[M]":
     NORMAL :
     ER = ER * (ER >3 AND ER <11)
        +  NOT (ER >3 AND ER <11):
     ON ER GOTO 182:
```

```
     ON ER = 5 GOTO 182:
     ON ER >3 AND ER <11 GOTO 188

182  GOSUB 152:
     RESUME

184  END

186  PRINT "[G][G][M]SORRY, OUT OF MEMORY AT LINE ";
        PEEK (75) + 256 * PEEK (76):
     NORMAL :
     END

188  PRINT ER$(ER - 4):
     ON ER = 6 GOSUB 192:
     ON ER = 4 OR ER = 5 OR ER = 8 OR ER = 9 GOSUB 4:
     ON ER <10 GOTO 182:
     & I("UNLOCK? (Y/N)";"Y";A$,"9",0,1):
     ON A$ = "N" GOTO 182:
     PRINT "[M]UNLOCKING FILE":
     PRINT D$;"UNLOCK ";NA$;DR$(DR):
     GOTO 182

190  E2 = 1:
     POKE 216,0:
     ON PEEK (222) = 255 GOTO 184:
     ON PEEK (222) = 77 GOTO 186::
     PRINT CHR$ (7) CHR$ (7)"[M]
        DATA DISKETTE IS NOT IN DRIVE":
     GOSUB 4:
     RESUME

192  & I("[M]CATALOG? (Y/N)";"N";A$,"10",0,1):
     ON A$ = "N" GOTO 16:
     PRINT D$"CATALOG"DR$(DR):
     RETURN
```

(1,194-208) Cold start initialization.

```
194  HOME :
     INVERSE :
     FOR I = 1 TO 4:
        PRINT TAB( 2)" " TAB( 38)" ":
     NEXT I:
     NORMAL :
     VTAB 2:
     HTAB 3:
     PRINT " APPLIED FINITE ELEMENT ANALYSIS: ":
     HTAB 3:
     PRINT "     AN APPLE " CHR$ (221) CHR$ (219)
        " IMPLEMENTATION     ":
     VTAB 6:
     HTAB 12:
     PRINT "COPYRIGHT, 1985"

196  VTAB 8:
     INVERSE :
     HTAB 14:
     PRINT "SOLVE";:
     NORMAL :
     PRINT " (4.00)[M][M]ABSTRACT:[M][M]
        FINITE ELEMENT EQUATIONS ARE DEFINED FOREACH ELEMENT,
        ASSEMBLED INTO A GLOBAL[M]
        MATRIX EQUATION, MODIFIED BY BOUNDARY[M]CONDITIONS,
        AND SOLVED FOR NODE[M]TEMPERATURES."
```

```
198  GOSUB 4:
     & C(ER$,DR$):
     DIM ER$(6),DR$(2):
     ER$(0) = "DISKETTE IS WRITE PROTECTED":
     DR$(1) = ",D1":
     DR$(2) = ",D2":
     DR = PEEK (2048):
     ON DR = 2 GOTO 200:
     & B(200,20):
     PRINT "INSERT ";:
     FLASH :
     PRINT "DATA";:
     NORMAL :
     PRINT " DISKETTE INTO DRIVE":
     GOSUB 4

200  ER$(2) = "FILE NOT FOUND":
     ER$(3) = "VOLUME MISMATCH":
     ER$(4) = "I/O ERROR - DOOR/DISK INIT":
     ER$(5) = "DISK IS FULL":
     GOSUB 160:
     HOME :
     GOSUB 176:
     U$ = "----------------------------------------":
     PRINT U$:
     POKE 34,7:
     HOME

202  & C(SI%,EN%,NC,MP,GF,GS):
     DIM SI%(22):
     RE = 5:
     GOSUB 152:
     NE = N1:
     DE = N2:
     DK = DE:
     RE = 6:
     GOSUB 152:
     NN = N1:
     DIM EN%(NE,DE),NC(NN,2),MP(NE,3):
     GOSUB 144:
     ON BC% = 2 GOTO 184:
     BW = EN%(0,0):
     DD = 2:
     DIM GF(NN),GS(NN,BW):
     ONERR  GOTO 180

204  & C(PN$,CR$):
     DIM PN$(1),CR$(1):
     PN$(0) = "HELLO":
     PN$(1) = "POSTPROCESS":
     CR$(0) = " ":
     CR$(1) = CHR$ (13):
     DEF FN AR(EL) = .5 * (X(2) * Y(3) + X(3) * Y(1)
        + X(1) * Y(2) - X(2) * Y(1) - X(3) * Y(2)
        - X(1) * Y(3)):
     RE = 9:
     GOSUB 152

206  EF = 0:
     GOSUB 164:
     & R(MP,N$):
     ON EF GOTO 206:
     GOSUB 172:
     UN = (MP(0,1) * MP(0,2) * MP(0,3) = 1):
```

```
     RE = 11:
     GOSUB 152:
     NP = N1:
     & C(BC):
     DIM BC(NP,2):
     DEF  FN P1(I) = I - 1 +  DE * (I = 1):
     DEF  FN P2(I) = I +  1 - DE * (I = DE):
     PI =  ATN (1) * 4:

208  EF = 0:
     GOSUB 164:
     & R(BC,N$):
     ON EF GOTO 208:
     HB = BC(0,0):
     HF = BC(0,1):
     GOSUB 172:
     GOSUB 4
```

(2,210-218) Warm restart, menu, and branching.

```
210  FR = 0:
     TEXT :
     HOME :
     EO = 1:
     ON Z = 2 AND  PEEK (2166) > = 6 GOTO 222:
     NM =  PEEK (2166):
     MO = 1:
     FR = (Z = 1 OR Z = 2)

212  PRINT "*** PROBLEM SOLUTION (4.05) ***[M][M]
        --- FORM & SOLVE ALL EQUATIONS ---[M]
        1. DO COMPLETE PROBLEM SOLUTION (MENU[M]
           OPTIONS 2 - 5)[M][M]
        --- FORM SYSTEM EQUATIONS ---[M]
        2. ASSEMBLE GLOBAL FORCE (SOURCE) AND"

214  PRINT "   STIFFNESS (COEFFICIENT) MATRICES[M]
        3. APPLY CONVECTION BOUNDARY CONDITIONS[M]
        4. APPLY CONSTANT TEMPERATURE BOUNDARY    CONDITIONS[M][M]
        --- SOLVE SYSTEM EQUATIONS ---[M]
        5. SOLVE FOR NODE TEMPERATURES[M][M]
        --- LIST RESULTS ---"

216  PRINT "6. LIST NODE TEMPERATURES[M][M]
        0. NONE OF ABOVE[M]":
     ON Z = 2 GOTO 218:
     & I(SR$,NM;MO,"11",BC%,1,MO >
        = 0 AND MO <7)

218  ON MO = 0 GOTO 220:
     FOR RE = 12 TO 16:
         SI%(RE) = 0:
     NEXT RE:
     ON MO GOSUB 230,40,64,86,114,134:
     GOTO 210
```

(220-226) Exit menu and actions.

```
220  HOME :
     HTAB 10:
     PRINT "* SOLVE: EXIT (4.7)*[M][M]
        1. PROCEED TO 'POSTPROCESS'[M]
        2. EXIT TO MAIN MENU[M]
        3. DON'T EXIT; REMAIN IN 'SOLVE'[M][M]
        0. STOP[M][M]":
     & I(SR$,1;EO,"12",BC%,1,EO > = 0 AND EO <4):
```

```
     ON BC% = 2 OR EO = 0 GOTO 184:
     ON EO = 3 GOTO 210

222  IF DR = 1 THEN
       & B(200,20):
       PRINT "[M]INSERT ";:
       FLASH :
       PRINT "PROGRAM";:
       NORMAL :
       PRINT " DISKETTE CONTAINING[M]'";:
       INVERSE :
       PRINT PN$(EO = 1);:
       NORMAL :
       PRINT "' INTO DRIVE":
       FR = 0:
       GOSUB 4:
       ON BC% = 1 OR BC% = 3 GOTO 220

224  IF EO = 2 THEN
       POKE 103,1:
       POKE 104,64:
       POKE 16384,0

226  POKE 2166,1:
     PRINT D$"RUN "PN$(EO = 1)",D1"

228  & B(10,10):
     & B(0,0,10):
     RETURN
```

(230-234) Menu option 1: Execute options 2 through 4 (assemble the global force and stiffness matrices, apply convection boundary conditions, apply constant temperature boundary conditions and solve for nodal temperatures).

```
230  ON Z = 2 GOTO 232:
     POKE 34,0:
     HOME :
     PRINT "* DO COMPLETE PROBLEM SOLUTION (4.1) *[M][M]
        OPTIONS:[M][M]
        1. RUN W/O USER INTERACTION[M]    (USING ALL DEFAULTS)[M]
        2. RUN WITH USER INTERACTION[M][M]":
     & I(SR$,1;FR,"13",BC%,1,FR >0 AND FR <3):
     ON BC% < >0 GOTO 16:
     FR = (FR = 1)

232  HOME :
     PRINT "DOING MENU OPTION 2:[M][M]
        ASSEMBLY OF GLOBAL FORCE AND STIFFNESS  MATRICES":
     GOSUB 228:
     GOSUB 40:
     HOME :
     PRINT "DOING MENU OPTION 3:[M]
        APPLICATION OF CONVECTION BOUNDARY[M]  CONDITIONS":
     GOSUB 228:
     GOSUB 64:
     HOME :
     PRINT "DOING MENU OPTION 4:"

234  HOME :
     PRINT "[M]  APPLICATION OF CONSTANT
        TEMPERATURE   BOUNDARY CONDITIONS":
     GOSUB 228:
     GOSUB 86:
     HOME :
```

```
        PRINT "DOING MENU OPTION 5:[M][M]
           SOLVING FOR NODE TEMPERATURES":
        GOSUB 228:
        GOSUB 114:
        ON FR = 0 GOTO 16:
        & B(50,20,3):
        RETURN

65535 REM [M][M]1FEB85[M]JRC/DCD

END-OF-LISTING
```

POSTPROCESS.HEAT

(1,158-174) Cold start initialization.

```
1  TEXT :
   & D(T):
   CLEAR :
   P% = 10:
   & L(255):
   D$ = CHR$ (4):
   SR$ = "... SELECT BY NUMBER <RETURN>":
   FI$ = "FILEINFO.TXT":
   POKE 2164,5:
   Z = PEEK (2163):
   GOTO 158:
   REM POST(H)
```

(2,176-182) Warm restart (menu and branching).

```
2  GOTO 176
```

Wait for user response, clear screen, and continue.

```
4  ON FR = 1 GOSUB 190:
   VTAB (FR = 1) + 24 * (FR = 0):
   CALL - 958:
   ON FR = 1 GOTO 10:
   & B(20,20):
   & I("... WHEN READY,
     PRESS <RETURN>";"";A$,"1",BC%):
   HOME :
   RETURN
```

(6-10) Decompose the conjugate global equations using the Gauss Elimination Method.

```
6  FOR I = 1 TO NN - 1:
     VTAB PEEK (37):
     PRINT " WORKING ON ROW #";I;" OF ";NN:
     C1 = CS(I,1):
     MJ = I + BW - 1:
     MJ = MJ * (MJ < = NN)
        + NN * (MJ >NN):
     N = 0:
     MK = BW:
     IF NN - I + 1 <BW THEN
        MK = NN - I + 1

8    FOR J = I + 1 TO MJ:
        MK = MK - 1:
        N = N + 1:
        C2 = CS(I,N + 1) / C1:
        FOR K = 1 TO MK:
           CS(J,K) = CS(J,K) - C2 * CS(I,N + K):
        NEXT K:
        FOR L = 1 TO DD:
           RH(J,L) = RH(J,L) - C2 * RH(I,L):
   NEXT L,J,I:
   VTAB PEEK (37):
   CALL - 868

10 RETURN
```

Print temperature of element EL.

```
12 PRINT " ELEMENT "EL" TEMPERATURE IS "TE(EL)"[M]":
   RETURN
```

Define local node numbers, nodal coordinates, and area for element EL.

```
14  FOR I = 1 TO DE:
        ND(I) = EN%(EL,I):
        X(I) = NC(ND(I),1):
        Y(I) = NC(ND(I),2):
    NEXT I:
    A2 = X(2) * Y(3) +  X(3) * Y(1) +  X(1) * Y(2)
       - X(2) * Y(1) - X(3) * Y(2) - X(1) * Y(3):
    RETURN
```

Define shape function gradient matrix for element EL.

```
16  & M(B = (0)):
    FOR J = 1 TO DE:
        I =  FN P1(J):
        K =  FN P2(J):
        B(1,J) = (Y(K) - Y(I)) / A2:
        B(2,J) = (X(I) - X(K)) / A2:
    NEXT J:
    RETURN
```

(18-28) Menu option 2: Solve for element temperatures.

```
18  POKE 34,0:
    HOME :
    PRINT "* SOLVE FOR ELEMENT TEMPERATURES (5.2) *":
    POKE 34,2:
    RE = 17:
    & C(UU,U,TE):
    GOSUB 126:
    GOSUB 118:
    ON SI%(RE) = 0 GOTO 200:
    DIM UU(NN),TE(NE):
    ONERR  GOTO 144
```

If necessary, retrieve nodal temperatures (RE=17).

```
20  EF = 0:
    GOSUB 130:
    & R(UU,N$):
    GOSUB 138:
    GOSUB 4:
    HOME :
    PO = 0:
    ON FR GOTO 22:
    & I("DISPLAY ELEMENT TEMPERATURES AS OBTAINED";
       "N";A$,"2",BC%,1):
    ON BC% GOTO 20,10,20:
    PO = (A$ = "Y")


22  PRINT "[M]SOLVING FOR ELEMENT TEMPERATURES...":
    & L(P%):
    FOR EL = 1 TO NE:
        TE(EL) = 0:
        FOR I = 1 TO DE:
            TE(EL) = TE(EL) +  UU(EN%(EL,I)):
        NEXT I:
        TE(EL) = TE(EL) / DE:
        PRINT ".";:
        ON PO GOSUB 12:
    NEXT EL:
    PRINT :
    & L(255):
    GOSUB 4
```

```
24  HOME :
    RE = 18:
    GOSUB 126:
    GOSUB 118:
    N1 = NE:
    N2 = 1:
    ONERR  GOTO 144

26  EF = 0:
    GOSUB 120:
    ON  PEEK (2163) = 3 GOTO 28:
    & S(TE,N$)

28  ON EF GOTO 26:
    GOSUB 122:
    POKE 2166,3:
    GOTO 4
```

(30-46) Menu option 3: Find element temperature gradients.

```
30  POKE 34,0:
    HOME :
    PRINT "* SOLVE FOR ELEM TEMP GRADIENTS  (5.3) *":
    POKE 34,2:
    HOME :
    RE = 17:
    & C(UU,U,B,TG):
    GOSUB 126:
    GOSUB 118:
    ON SI%(RE) = 0 GOTO 200:
    DIM UU(NN),U(DK,1),B(DD,DK),TG(NE,DD +  2):
    ONERR  GOTO 144
```

If necessary, retrieve nodal temperatures (RE=17).

```
32  EF = 0:
    GOSUB 130:
    & R(UU,N$):
    ON EF GOTO 32:
    GOSUB 138:
    GOSUB 4:
    HOME :
    PO = 0:
    ON FR GOTO 34:
    GOSUB 44:
    ON BC% GOTO 32,10,32:
    RW = 2:
    NR = NE:
    A = 0
```

(34-36) Define element temperature gradient components, magnitude, and direction.

```
34  HOME :
    PRINT "CALCULATING ELEMENT TEMPERATURE GRADIENT":
    FOR EL = 1 TO NE:
        VTAB 4:
        PRINT " FOR ELEMENT "EL" OF "NE:
        GOSUB 14:
        GOSUB 16:
        FOR I = 1 TO DE:
            U(I,1) = UU(ND(I)):
        NEXT I

36      FOR I = 1 TO DD:
            TG(EL,I) = 0:
            FOR K = 1 TO DK:
```

```
            TG(EL,I) = TG(EL,I) +  B(I,K) * U(K,1):
        NEXT K,I:
        X1 = TG(EL,1):
        Y1 = TG(EL,2):
        TG(EL,DD +  1) =  SQR (X1 ^ 2 +  Y1 ^ 2):
        TG(EL,DD +  2) =  FN AN(X1):
     NEXT EL:
     ON PO >0 GOSUB 46:
     GOSUB 4
```

(38-40) Save element temperature gradients (RE=19).

```
38   HOME :
     RE = 19:
     GOSUB 126:
     GOSUB 118:
     N1 = NE:
     N2 = DD +  2:
     ONERR  GOTO 144

40   EF = 0:
     GOSUB 120:
     ON  PEEK (2163) = 3 GOTO 42:
     & S(TG,N$)

42   ON EF GOTO 44:
     GOSUB 122:
     & C(TG):
     POKE 2166,4:
     GOTO 4

44   PRINT "[M]GRADIENT DISPLAY OPTIONS[M][M]
        1. "DI$(1)" AND "DI$(2)" COMPONENTS[M]
        2. RESULTANT (MAGNITUDE & ANGLE)[M] [M]
        0. NONE[M]":
     & I(SR$,1;PO,"3",BC%,1,PO  = 0 AND PO <3):
     RETURN
```

List temperature gradient.

```
46   HOME :
     & L(P%):
     INVERSE :
     PRINT  TAB( 16)VA$(2) TAB( 39)" [M]"
        RW$(RW) TAB( 10)LA$(PO) TAB( 39)" ":
     NORMAL :
     POKE 34,4:
     FOR I = 1 TO NR:
         PRINT " "I TAB( 9)TG(I,1 +  2
            * (PO = 2)) TAB( 25)TG(I,2 +  2 * (PO = 2)):
     NEXT I:
     & L(255):
     RETURN
```

(48-68) Menu option 4: Solve for nodal temperature gradients (load element temperature gradients, form global conjugate matrices, decompose and solve equations for nodal temperature gradients, and save (RE=20).
(RE=20).

```
48   POKE 34,0:
     HOME :
     PRINT "* SOLVE FOR NODE TEMP. GRADIENTS (5.4) *":
     POKE 34,2:
     RE = 19:
     & C(CE,RE,CS,RH,EG):
     GOSUB 126:
     GOSUB 118:
```

```
    ON SI%(RE) = 0 GOTO 200:
    DIM CE(DE,DE),RE(DE,DD),CS(NN,BW),RH(NN,DD),EG(N1,N2):
    ONERR  GOTO 144

50  EF = 0:
    GOSUB 130:
    & R(EG,N$):
    ON EF GOTO 50:
    GOSUB 138:
    GOSUB 4:
    HOME :
    PO = 0:
    ON FR GOTO 52:
    & I("DISPLAY CONJUGATE MATRICES";"N";A$,"4",BC%,1):
    ON BC% GOTO 48,10,48:
    PO = (A$ = "Y")
```

Define element conjugate matrices.

```
52  HOME :
    PRINT "FORMING CONJUGATE MATRICES":
    POKE 34,3:
    FOR EL = 1 TO NE:
        VTAB 4:
        CALL  - 958:
        PRINT " ELEMENT "EL" OF "NE:
        GOSUB 14:
        C1 = A2 / 24:
        & M(CE = (C1)):
        C2 = 2 * C1:
        FOR I = 1 TO DE:
            CE(I,I) = C2:
            FOR J = 1 TO DD:
                RE(I,J) = EG(EL,J):
        NEXT J,I:
        & M(RE = RE * (4 * C1)):
        ON PO = 0 GOTO 56

54      & L(P%):
        PRINT "[M]ELEMENT CONJ. STIFFNESS":
        INVERSE :
        PRINT "COLUMNS:  ";:
        FOR J = 1 TO DE:
            PRINT EN%(EL,J)"    ";:
        NEXT J:
        PRINT  TAB( 39);" ":
        NORMAL :
        FOR I = 1 TO DE:
            PRINT "ROW "EN%(EL,I)":";:
            FOR J = 1 TO DE:
                VTAB 8 +  (J >2) +  2 * (I - 1):
                HTAB 10 +  16 * (J = 2):
                PRINT CE(I,J):
        NEXT J,I:
        PRINT
```

Assemble element conjugate matrices into global conjugate matrices.

```
56      FOR I = 1 TO DE:
            KI = EN%(EL,I):
            FOR L = 1 TO DD:
                RH(KI,L) = RH(KI,L) +  RE(I,L):
            NEXT L:
            FOR J = 1 TO DE:
                KJ = EN%(EL,J):
                IF KJ > = KI THEN
```

```
                     K = KJ - KI + 1:
                     CS(KI,K) = CS(KI,K) + CE(I,J):
                     IF PO = 1 THEN
                          PRINT "UPDATED GLOBAL K("KI",
                             "KJ") IS "CS(KI,K)

58          NEXT J,I:
            PRINT :
            & L(255):
      NEXT EL:
      & C(EG):
      SI%(19) = 0:
      POKE 34,2:
      HOME :
      PRINT "*** CONJUGATE MATRICES ASSEMBLED ***";
      ON FR GOSUB 190:
      PO = 0:
      ON FR GOTO 60:
      GOSUB 44:
      ON BC% GOTO 10,10:
      RW = 1:
      NR = NN:
      A = 0

60    HOME :
      PRINT "DECOMPOSING MATRICES...[M]":
      GOSUB 6:
      HOME :
      PRINT "[M]SOLVING FOR NODE TEMPERATURE GRADIENTS";
      & C(TG):
      DIM TG(NN,DD + 2)
```

Solve for nodal temperature gradients by Gauss elimination.

```
62    FOR L = 1 TO DD:
          TG(NN,L) = RH(NN,L) / CS(NN,1):
          FOR K = 1 TO NN - 1:
              PRINT ".";
              I = NN - K:
              MJ = BW * (I + BW - 1 < = NN)
                 + (NN - I + 1) * (I + BW - 1 >NN):
              SU = 0:
              FOR J = 2 TO MJ:
                  N = I + J - 1:
                  SU = SU + CS(I,J) * TG(N,L):
              NEXT J:
              TG(I,L) = (RH(I,L) - SU) / CS(I,1):
      NEXT K,L:
      PRINT :
      & C(CS,CE,RH,RE)
```

Define gradient vector direction.

```
64    FOR I = 1 TO NN:
          X1 = TG(I,1):
          Y1 = TG(I,2):
          TG(I,DD + 1) = SQR (X1 ^ 2 + Y1 ^ 2):
          TG(I,DD + 2) = FN AN(X1):
      NEXT I:
      ON PO >0 GOSUB 46:
      POKE 34,2:
      GOSUB 4:
      HOME :
      RE = 20:
      GOSUB 126:
      GOSUB 118:
```

```
    N1 = NN:
    N2 = DD +  2:
    ONERR  GOTO 144

66  EF = 0:
    GOSUB 120:
    ON  PEEK (2163) = 3 GOTO 68:
    & S(TG,N$)

68  ON EF GOTO 66:
    GOSUB 122:
    & C(TG):
    POKE 34,0:
    POKE 2166,5:
    GOSUB 4:
    RETURN
```

(70-80) Menu option 7: List results (load temperatures, temperature gradients, resultant heat sources, or surface heat fluxes and list them).

```
70  POKE 34,0:
    HOME :
    PRINT "  LIST RESULTS  (5.7)[M][M]":
    POKE 34,2:
    ON L2 = 0 GOSUB 126:
    PRINT "LISTING OPTIONS:[M]":
    FOR I = 0 TO 5:
        PRINT I + 1". "MU$(I):
    NEXT I:
    PRINT "[M[O. NONE OF THESE[M]":
    & I(SR$,0;LO "5",BC%,1,LO > = 0 AND LO <7):
    L2 = 1:
    ON BC% < >0 OR LO = 0 GOTO 10

72  HOME :
    INVERSE :
    L$ = MU$(LO - 1):
    PRINT "LIST "L$:
    NORMAL :
    PRINT :
    RE = 16 +  LO:
    GOSUB 118:
    ON SI%(RE) = 0 GOTO 200:
    VA = (LO = 1 OR LO = 2) +  2 * (LO = ERR  GOTO 144

74  IF N2 = 1 THEN
        & C(TG):
      DIM TG(N1)

76  EF = 0:
    GOSUB 130:
    & R(TG,N$):
    ON EF GOTO 76:
    GOSUB 138:
    GOSUB 4:
    PRINT L$":[M]":
    RW = 1 +  (LO = 2 OR LO = 3):
    I = 1 +  (LO = 3 OR LO = 4) +  2 * (LO = 6):
    ON I = 2 GOSUB 44:
    NR = N1:
    ON I GOSUB 78,46,80:
    & C(TG):
    FR = 0:
    GOSUB 4:
    GOTO 70
```

List temperatures or equivalent heat sources.

```
78  & L(P%):
    INVERSE :
    PRINT RW$(RW) TAB( 12)VA$(VA) TAB( 39)" ":
    NORMAL :
    POKE 34, PEEK (37):
    FOR I = 1 TO N1:
        PRINT " "I TAB( 12)TG(I):
    NEXT I:
    & L(255):
    RETURN
```

List heat fluxes across boundary lines.

```
80  & L(P%):
    INVERSE :
    PRINT "NODES       FLUX (OUT)" TAB( 39)" ":
    NORMAL :
    POKE 34, PEEK (37):
    FOR I = 1 TO N1:
        PRINT TG(I,0)"-"TG(I,1) TAB( 12)TG(I,2):
    NEXT I:
    & L(255):
    RETURN
```

(82-96) Menu option 5: Evaluate nodal heat sources.

```
82  POKE 34,0:
    HOME :
    PRINT "* EVALUATE NODE HEAT SOURCES (5.5) *[M][M]":
    POKE 34,2:
    RE = 13:
    GOSUB 126:
    GOSUB 118:
    ON SI%(RE) = 0 GOTO 200:
    NN = N1:
    BW = N2:
    & C(GS,TE,RH):
    DIM GS(NN,BW),TE(NN),RH(NN):
    ONERR  GOTO 144
```

If necessary, retrieve the initial global stiffness matrix (RE=13).

```
84  EF = 0:
    GOSUB 130:
    & R(GS,N$):
    ON EF GOTO 84:
    GOSUB 138:
    RE = 17:
    GOSUB 118:
    ON SI%(RE) = 0 GOTO 200
```

If necessary, retrieve nodal temperatures (RE=17).

```
86  EF = 0:
    GOSUB 130:
    & R(TE,N$):
    ON EF GOTO 86:
    GOSUB 138:
    GOSUB 4:
    W = 1:
    Z5 = 0:
    HOME :
    FOR I = 1 TO NN:
        VTAB 3:
        PRINT "EVALUATING ROW "I" OF "NN
```

(88-90) Calculate resultant nodal heat sources and list them.

```
88      SU = Z5:
        K = I - W:
        FOR J = 2 TO BW:
            M = J +  I - W:
            RW = (M < = NN) +  2 * (K >Z5):
            IF RW < >Z5 THEN
                I1 = I * (RW < >2) +  K * (RW = 2):
                M1 = M * (RW < >2) +  K * (RW = 2):
                SU = SU +  GS(I1,J) * TE(M1):
                K = K - (RW = 2):
                IF RW = 3 THEN
                    SU = SU +  GS(K,J) * TE(K):
                    K = K - W

90      NEXT J:
        RH(I) = SU +  GS(I,W) * TE(I):
    NEXT I:
    HOME :
    ON FR GOTO 92:
    & L(P%):
    INVERSE :
    PRINT "NODE     HEAT SOURCES" TAB( 39)"":
    NORMAL :
    POKE 34,3:
    FOR I = 1 TO NN:
        PRINT " "I TAB( 9)RH(I):
    NEXT I:
    & L(255)
```

(92-94) Save nodal equivalent heat sources (RE=21).

```
92  POKE 34,2:
    ON FR = 0 GOSUB 4:
    RE = 21:
    GOSUB 118:
    N1 = NN:
    N2 = 1:
    ONERR  GOTO 144

94  EF = 0:
    GOSUB 120:
    ON  PEEK (2163) = 3 GOTO 96:
    & S(RH,N$)

96  ON EF GOTO 94:
    GOSUB 122:
    & C(GS,TE,RH):
    POKE 2166,6:
    GOTO 4
```

(98-110) Menu option 6: Evaluate surface fluxes (load boundary nodes and element temperature gradients, calculate heat fluxes across boundary lines and save heat fluxes (RE=22).

```
98  POKE 34,0:
    HOME :
    PRINT "* EVALUATE SURFACE FLUXES (5.6) *[M][M]":
    POKE 34,2:
    RE = 8:
    GOSUB 126:
    GOSUB 118:
    NB = N1:
    ON SI%(RE) = 0 GOTO 200:
    & C(BN%,TG,BF,IS):
    DIM BN%(NB,2),TG(NE,DD +  2),BF(NB,2),IS(10):
    ONERR  GOTO 144
```

Retrieve renumbered boundary nodes and elements and temperature (RE=8) gradients at the elements (RE=19).

```
100  EF = 0:
     GOSUB 130:
     & R(BN%,N$):
     ON EF GOTO 100:
     GOSUB 138:
     NS = BN%(0,0):
     FOR I = 1 TO 2 * NS:
         IS(I) = BN%(I,0):
     NEXT I:
     RE = 19:
     GOSUB 126:
     GOSUB 118:
     ON SI%(RE) = 0 GOTO 200:
     EF = 0:
     GOSUB 130:
     & R(TG,N$):
     ON EF GOTO 100:
     GOSUB 138:
     GOSUB 4
```

Calculate heat flux across boundary line segments

```
102  PRINT "CALCULATING HEAT FLUXES ";:
     N = 0:
     FOR B = 1 TO NS:
         I1 = IS(2 * B - 1):
         I2 = IS(2 * B):
         FOR I = I1 TO I2:
             N = N + 1

104          PRINT ".";:
             J = (I + 1) * (I <I2) + I1 * (I = I2):
             N1 = BN%(I,1):
             X1 = NC(N1,1):
             Y1 = NC(N1,2):
             N2 = BN%(J,1):
             X2 = NC(N2,1):
             Y2 = NC(N2,2):
             L = FN LL(X1):
             EL = BN%(I,2):
             BF(N,0) = N1:
             BF(N,1) = N2:
             BF(N,2) = - MP(EL,1) * TG(EL,1)
                * (Y2 - Y1) / L - MP(EL,2) * TG(EL,2)
                * (X1 - X2) / L:
     NEXT I,B:
     PRINT
```

(106-108) Save heat fluxes (RE=22).

```
106  HOME :
     RE = 22:
     GOSUB 118:
     N1 = NB:
     N2 = 2

108  EF = 0:
     GOSUB 120:
     ON PEEK (2163) = 3 GOTO 110:
     & S(BF,N$)

110  ON EF GOTO 108:
     GOSUB 122:
```

```
    POKE 34,0:
    GOSUB 4:
    & C(BF,BN%,IS,TG):
    POKE 2165,6:
    POKE 2166,7:
    RETURN
```

(112-116) Load element nodal numbers and nodal coordinates (RE=5,6).

```
112 RE = 5:
    GOSUB 126:
    GOSUB 118:
    ON SI%(RE) = 0 GOTO 200:
    ONERR  GOTO 144

114 EF = 0:
    GOSUB 130:
    & R(EN%,N$):
    ON EF GOTO 114:
    GOSUB 138:
    RE = 6:
    GOSUB 118:
    ON SI%(RE) = 0 GOTO 200

116 EF = 0:
    GOSUB 130:
    & R(NC,N$):
    ON EF GOTO 116:
    GOSUB 138:
    RETURN
```

Retrieve information about record RE.

```
118 N1 = FRE (0):
    PRINT D$"OPEN"FI$",L100"DR$(DR):
    PRINT D$"READ"FI$",R"RE:
    INPUT SI%(RE),NA$,DE$,N1,D1$,N2,D2$:
    PRINT D$"CLOSE"FI$:
    RETURN
```

Request filename for data to be saved.

```
120 ON Z = 2 GOTO 134:
    PRINT "[M]*** NOW ";:
    INVERSE :
    PRINT "SAVING";:
    NORMAL :
    PRINT " DATA TO DISKETTE ***":
    ON FR GOTO 132:
    ON Z GOTO 136:
    & I("[M]FILE NAME FOR " + DE$,NA$,NA$,"6",BC%,30):
    ON BC% < >0 GOTO 120:
    GOTO 132
```

Set active status and save array descriptors for record RE.

```
122 SI%(RE) = 1:
    ON  PEEK (2163) = 3 GOTO 10:
    PRINT D$"OPEN"FI$",L100"DR$(DR):
    PRINT D$"WRITE"FI$",R"RE:
    PRINT SI%(RE):
    PRINT NA$:
    PRINT DE$:
    PRINT N1:
    PRINT D1$:
    PRINT N2:
    PRINT D2$:
    PRINT D$"CLOSE"FI$
```

Print record information.

```
124  ON Z = 2 GOTO 10:
     PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2" "D2$"[M]
        WERE SAVED"CR$(FR):
     ON FR = 0 GOTO 10:
     PRINT "IN FILE:";:
     INVERSE :
     PRINT NA$:
     NORMAL :
     RETURN
```

(126-128) Verify existence of data files.

```
126  ONERR  GOTO 152

128  E2 = 0:
     PRINT D$:
     PRINT D$"CKFILE"FI$;DR$(DR):
     ON E2 GOTO 128:
     RETURN
```

(130-136) Request filename for data to be loaded.

```
130  ON Z = 2 GOTO 134:
     PRINT "[M]*** ";:
     INVERSE :
     PRINT "LOADING";:
     NORMAL :
     PRINT " DATA FROM DISKETTE ***":
     ON FR GOTO 132:
     ON Z GOTO 136:
     & I("[M]FILE NAME FOR[M]" +  DE$,NA$,NA$,"7",BC%,30):
     ON BC% < >0 GOTO 130

132  ON  LEFT$ (NA$, LEN (KW$)) = KW$ GOTO 134:
     NA$ = KW$ +  "/" +  NA$

134  N$ = NA$ +  DR$(DR):
     RETURN

136  PRINT " ";NA$:
     GOTO 132
```

Print data descriptors.

```
138  ON Z = 2 GOTO 10:
     PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2" "D2$"[M]
        WERE LOADED"CR$(FR):
     ON FR = 0 GOTO 10:
     PRINT "FROM FILE:";:
     INVERSE :
     PRINT NA$:
     NORMAL :
     RETURN
```

Read problem description.

```
140  PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"READ"FI$",R0":
     INPUT TE,KW$,PD$:
     PRINT D$"CLOSE"FI$:
     RETURN
```

Print problem description.

```
142  ON LEN (KW$) = 0 GOSUB 140:
     PRINT "[M]PROBLEM KEYWORD: "KW$"[M][M]
        PROBLEM DESCRIPTION:[M]"PD$:
     RETURN
```

(144-156) Trap errors.

```
144  EF = 1:
     ER =  PEEK (222):
     POKE 216,0:
     ON ER = 255 GOTO 148:
     FLASH :
     ON ER = 77 GOTO 156:
     PRINT "[M]" CHR$ (7) CHR$ (7)"ERROR "ER"[M]":
     NORMAL :
     ER = ER * (ER >3 AND ER <11)
        +   NOT (ER >3 AND ER <11):
     ON ER GOTO 146:
     ON ER = 5 GOTO 146:
     ON ER >3 AND ER <11 GOTO 150

146  GOSUB 118:
     RESUME

148  END :
     GOTO 2

150  PRINT ER$(ER - 4):
     ON ER = 6 GOSUB 154:
     ON ER = 4 OR ER = 5 OR ER = 8 OR ER = 9 GOSUB 4:
     ON ER <10 GOTO 146:
     & I("UNLOCK? (Y/N)";"Y";A$,"8",0,1):
     ON A$ = "N" GOTO 146:
     PRINT D$"UNLOCK "NA$DR$(DR):
     GOTO 146

152  E2 = 1:
     ON  PEEK (222) = 255 GOTO 148:
     POKE 216,0:
     ON ER = 77 GOTO 156:
     PRINT "[M]" CHR$ (7) CHR$ (7)
        "DATA DISKETTE IS NOT IN DRIVE":
     GOSUB 4:
     RESUME

154  & I("[M]CATALOG? (Y/N)";"N";A$,"9",0,1):
     ON A$ = "N" GOTO 10:
     PRINT D$"CATALOG"DR$(DR):
     RETURN

156  PRINT "[G][G][M]SORRY, OUT OF MEMORY AT LINE ";
        PEEK (75) +  256 *  PEEK (76):
     NORMAL :
     END
```

(1,158-174) Cold start initialization.

```
158  HOME :
     INVERSE :
     FOR I = 1 TO 4:
        PRINT  TAB( 2)" " TAB( 38)" ":
     NEXT I:
     NORMAL :
     VTAB 2:
     HTAB 3:
     PRINT " APPLIED FINITE ELEMENT ANALYSIS: ":
     HTAB 3:
     PRINT "    AN APPLE " CHR$ (221) CHR$ (219)
       " IMPLEMENTATION     [M][M][M]" TAB( 12)
       "COPYRIGHT, 1985[M]":
```

```
     HTAB 11:
     INVERSE

160  PRINT "POSTPROCESS";:
     NORMAL :
     PRINT " (5.0)[M][M]ABSTRACT:[M][M]
        NODE TEMPERATURES DETERMINED IN 'SOLVE' ARE
        USED TO DEFINE ELEMENT TEMPERATURES,TEMPERATURE
        GRADIENTS, NODE HEAT SOURCESAND HEAT FLUXES.  ANY
        OR ALL OF THESE[M]MAY BE DETERMINED.":
     GOSUB 4

162  & C(DR$,ER$,SI%,EN%,NC,MP):
     DIM DR$(2),ER$(6),SI%(22):
     DR$(1) = ",D1":
     DR$(2) = ",D2":
     DR =  PEEK (2048):
     ON DR = 2 GOTO 164:
     & B(200,20):
     PRINT "INSERT ";:
     FLASH :
     PRINT "DATA";:
     NORMAL :
     PRINT " DISKETTE INTO DRIVE":
     GOSUB 4

164  ER$(0) = "DISKETTE IS WRITE PROTECTED":
     ER$(2) = "FILE NOT FOUND":
     ER$(3) = "VOLUME MISMATCH":
     ER$(4) = "I/O ERROR - DOOR OPEN OR DISK NOT[M]
        INITIALIZED":
     ER$(5) = "DISK IS FULL":
     ER$(6) = "FILE IS LOCKED":
     & C(DF$,DI$,CR$):
     DIM DF$(4),DI$(2),CR$(1)

166  GOSUB 126:
     HOME :
     GOSUB 142:
     FOR I = 1 TO 38:
        PRINT "-";:
     NEXT :
     PRINT :
     POKE 34,7:
     HOME :
     RE = 5:
     GOSUB 118:
     NE = N1:
     DE = N2:
     DK = DE:
     RE = 6:
     GOSUB 118:
     NN = N1:
     DIM EN%(NE,DE),NC(NN,2),MP(NE,3):
     GOSUB 112:
     BW = EN%(0,0):
     DD = 2:
     & C(PN$,VA$,RW$):
     DIM PN$(1),VA$(3)

168  DF$(1) = "X":
     DF$(2) = "Y":
     DF$(3) = "R":
```

```
DF$(4) = "Z":
DI$(1) = DF$(2 * TE - 1):
DI$(2) = DF$(2 * TE):
CR$(0) = "[M]":
CR$(1) = " ":
PN$(0) = "HELLO":
PN$(1) = "PLOT":
& C(LA$,MU$):
DIM RW$(2),LA$(2),MU$(5)

170 GOSUB 192:
UN = (MP(0,1) * MP(0,2) * MP(0,3) = 1):
DEF FN P1(I) = I - 1 + DE * (I = 1):
DEF FN P2(I) = I + 1 - DE * (I = DE):
VA$(1) = "TEMPERATURE":
VA$(2) = "GRADIENT":
VA$(3) = "SOURCE":
RW$(1) = "NODE":
RW$(2) = "ELEMENT"

172 DEF FN LL(X1) = SQR ((X2 - X1) ^ 2 + (Y2 - Y1) ^ 2):
DEF FN AN(X1) = (90 * (Y1 > = 0) - 90 * (Y1 <0))
   * (X1 = 0) + (X1 < >0) * (180 * (X1 <0)
   + SGN (X1) * SGN (Y1) * RD * ATN ( ABS (Y1 / (X1 + (X1 = 0))))):
RD = 45 / ATN (1):
LA$(1) = "DT/D" + DI$(1) + "            DT/D" + DI$(2):
LA$(2) = "MAGNITUDE     ANGLE (DEG)"

174 MU$(0) = "NODE TEMPERATURES":
MU$(1) = "ELEMENT TEMPERATURES":
MU$(2) = "ELEMENT TEMPERATURE GRADIENTS":
MU$(3) = "NODE TEMPERATURE GRADIENTS":
MU$(4) = "RESULTANT NODE HEAT SOURCES":
MU$(5) = "RESULTANT SURFACE HEAT FLUXES"
```

(2,176-182) Warm restart (menu).

```
176 FR = 0:
NM = PEEK (2166):
TEXT :
HOME :
EO = 1:
ON Z = 2 AND PEEK (2165) = 6 GOTO 186

178 HTAB 7:
PRINT "* POST-PROCESSING (5.05) *[M]
   - SOLVE ALL EQUATIONS -[M]
   1. DO ALL CALCULATIONS (MENU OPTIONS     2 - 6)[M][M]
   - SOLVE FOR TEMPERATURES & GRADIENTS -[M]
   2. ";MU$(1);"[M]
   3. ";MU$(2);"[M]
   4. ";MU$(3)

180 PRINT "[M] - SOLVE FOR SOURCES & FLUXES -[M]
   5. ";MU$(4);"[M]
   6. ";MU$(5);"[M][M] - LISTING -[M]
   7. LIST OUTPUT[M][M]
   0. NONE OF THESE[M]"

182 FR = (Z = 1 OR Z = 2):
MO = 2:
ON Z = 2 GOTO 198:
& I(SR$,NM;MO,"10",BC%,1,MO > = 0 AND MO <8):
ON BC% GOTO 1,184,176:
L2 = 0:
```

```
ON MO + 1 GOTO 184,196:
ON MO - 1 GOSUB 18,30,48,82,98,70:
GOTO 176
```

(184-188) Exit menu.

```
184 HOME :
    HTAB 6:
    PRINT "* POSTPROCESS: EXIT  (5.8) *[M][M]
      1. PROCEED TO 'PLOT'[M]
      2. EXIT TO MAIN MENU[M]
      3. DON'T EXIT; REMAIN IN 'POSTPROCESS'[M][M]
      0. STOP[M]":
    & I(SR$,1;EO,"11",BC%,1,EO > = 0 AND EO <4):
    ON BC% GOTO 184:
    ON EO = 3 GOTO 176:
    ON EO = 0 OR BC% = 2 GOTO 148

186 POKE 2166,1:
    IF DR = 1 THEN
        & B(200,20):
        PRINT "[M]INSERT ";:
        FLASH :
        PRINT "PROGRAM";:
        NORMAL :
        PRINT " DISKETTE CONTAINING[M]'";:
        INVERSE :
        PRINT PN$(EO = 1);:
        NORMAL :
        PRINT "' INTO DRIVE":
        FR = 0:
        GOSUB 4:
        ON BC% = 1 OR BC% = 3 GOTO 184

188 POKE 103,1:
    POKE 104,64:
    POKE 16384,0:
    PRINT D$"RUN "PN$(EO = 1)",D1"
```

Delay subroutine.

```
190 & B(0,0,20):
    RETURN
```

(192-194) Retrieve material properties (RE=9).

```
192 RE = 9:
    GOSUB 126:
    GOSUB 118:
    ONERR GOTO 144

194 EF = 0:
    GOSUB 130:
    & R(MP,NA$):
    ON EF GOTO 194:
    GOSUB 138:
    RETURN
```

(196-198) Menu option 1: Do all calculations in options 2 through 6 (calculate element temperature gradients, nodal temperatures, nodal temperature gradients, resultant node heat sources, and resultant surface heat fluxes).

```
196 POKE 34,0:
    HOME :
    PRINT "* DO ALL CALCULATIONS (5.1) *[M][M]OPTIONS:[M][M]
      1. RUN W/O USER INTERACTION[M]   (USING
    ALL DEFAULTS)[M]
      2. RUN WITH USER INTERACTION[M]":
```

```
    & I(SR$,1;FR,"12",BC%,1,FR >0 AND FR <3):
    ON BC% < >0 GOTO 176:
    FR = (FR = 1):
    MO = 2

198 POKE 34,0:
    HOME :
    PRINT "DOING MENU OPTION "MO"[M][M]
      --- CALCULATING:[M]" SPC( 9)MU$(MO - 1):
    GOSUB 190:
    ON MO - 1 GOSUB 18,30,48,82,98:
    ON BC% = 2 GOTO 176:
    MO = MO +  1:
    ON MO <7 GOTO 198:
    ON FR = 0 GOTO 176:
    & B(50,20,3):
    FR = 0:
    GOTO 176
```

Print error message.

```
200 FLASH :
    ON  PEEK (222) = 77 GOTO 156:
    PRINT "[M]" CHR$ (7) CHR$ (7)"DATA NOT DEFINED":
    NORMAL :
    PRINT "EITHER THE FILE":
    INVERSE :
    PRINT NA$:
    NORMAL :
    PRINT "DOESN'T EXIST OR THE DATA IS NOT VALID":
    GOSUB 4:
    BC% = 2:
    RETURN

202 SI%(RE) = 1:
    RETURN

65535 REM [M][M]1FEB85[M]JRC/DCD

END-OF-LISTING
```

PLOT.HEAT

(1,254-280) Cold start initialization.

```
1  CLEAR :
   TEXT :
   D$ =  CHR$ (4):
   WR$ = "WHEN READY, PRESS <RETURN>":
   SR$ = "... SELECT BY NUMBER <RETURN>":
   U$ = "-------------------------------------":
   M$ = "USE <I><J><K><M>TO MOVE ":
   S$ = "<S>TO SELECT ":
   FI$ = "FILEINFO.TXT":
   POKE 2164,6:
   GOTO 254:
   REM PLOT(H)
```

(2,282-284) Warm restart (menu).

```
2  GOTO 282
```

Wait for user response, clear screen, and continue.

```
4  PRINT :
   & B(20,20):
   VTAB 24:
   CALL  - 868:
   & I(WR$;"";A$,"1",BC%):
   HOME :
   RETURN
```

Determine step size for moving labels.

```
6  GET G$:
   & B(150,4):
   AS% =  ASC (G$):
   CT% = 1 +  6 * (AS% < = 0 OR AS% > = 27):
   AS% = AS% +  64 * (AS% >0 AND AS% <27):
   RETURN
```

Erase background for label L$.

```
8  HCOLOR= 0:
   X = XP +  3.5 *  LEN (L$)
     + XA * (3.5 *  LEN (L$) +  5):
   Y = X - 7 *  LEN (L$):
   XP = XP - (X - 278) * (X >278)
     + ( ABS (Y) +  1) * (Y < = 0):
   FOR I1 = 1 TO 9:
      HPLOT XP - 3.5 *  LEN (L$) - 1
        + XA * (3.5 *  LEN (L$) +  5),YP - 4.5
        + I1 - YA * 5 TO XP +  3.5 *  LEN (L$)
        + XA * (3.5 *  LEN (L$) +  5),YP - 4.5 +  I1 - YA * 5:
   NEXT I1:
   HCOLOR= 3
```

Draw label L$ on graphics screen.

```
10 X = XP +  3.5 *  LEN (L$) + XA * (3.5 *  LEN (L$) +  5):
   Y = X - 7 *  LEN (L$):
   XP = XP - (X - 278) * (X >278)
     + ( ABS (Y) +  1) * (Y < = 0):
   FOR I1 = 1 TO  LEN (L$):
      XDRAW  ASC ( MID$ (L$,I1,1)) - 31 AT XP - 3.5
        *  LEN (L$) +  7 * (I1 - 1) +  XA
```

```
           * (3.5 *  LEN (L$) +  5),YP +  3.
        5 - YA * 5:
     NEXT I1:
     RETURN
```

Draw a symbol and move it between nodes or elements to select.

```
12   XDRAW 66 AT XL(N),YL(N):
     GET G$:
     XDRAW 66 AT XL(N),YL(N):
     & B(150,4):
     A =  ASC (G$):
     N = N +   FN S(NO):
     ON A < >83 AND A < >81 GOTO 12:
     L$ = LA$(LA) +   STR$ ( ABS (I%(N))):
     XP = XL(N):
     YP = YL(N):
     ON A = 83 AND A%(N) = 0 GOSUB 10:
     ON A = 83 GOSUB 48:
     ON A < >81 GOTO 12:
     A = 0:
     RETURN
```

Draw '+' symbol at ends of plot contour line segments for each element crossed.

```
14   LO = 0:
     ON ZT >0 GOSUB 26:
     ON LO < >0 GOTO 24:
     SCALE= 2:
     FOR J = 1 TO 2:
        DRAW 64 AT EX(J),EY(J):
     NEXT J:
     SCALE= 1:
     GOSUB 34:
     ON PF >0 GOTO 24:
     CV(CL,0) = CL:
     CV(CL,1) = EX(1):
     CV(CL,2) = EY(1):
     PF = 1:
     RETURN
```

(16-18) Draw symbol or contour line and move between lines to select contour label; draw line or label.

```
16   XDRAW 66 AT CV(N,1),CV(N,2):
     GET G$:
     XDRAW 66 AT CV(N,1),CV(N,2):
     & B(150,4):
     A =  ASC (G$):
     N = N +   FN C(P):
     ON A < >83 AND A < >81 GOTO 16:
     CL = CV(N,0):
     CV = V0 +  CL * DV:
     L$(1) =  CHR$ (CL +  65):
     L$(2) =  STR$ (CV):
     L$ = L$(CO):
     XP = CV(N,1):
     YP = CV(N,2)

18   I%(N) = I%(N) * (A < >83 OR
        (A = 83 AND A%(N))) +  (A = 83 AND A%(N) = 0):
     ON A = 83 AND A%(N) = 0 GOSUB 10:
     ON A = 83 GOSUB 48:
     ON A < >81 GOTO 16:
     A = 0:
     RETURN
```

Locate contour line ends within an element when two lines lie on different sides of an element.

```
20  P = P + 1:
    FR = (CV - S1) / (S2 - S1 + (S2 = S1)):
    FOR J = 1 TO 2:
        CL(P,J) = NC(N1,J) + FR * (NC(N2,J) - NC(N1,J)):
    NEXT J:
    ON P = 1 GOTO 24:
    FOR J = 1 TO 2:
        EX(J) = CL(J + (P = 3),1):
        EY(J) = CL(J + (P = 3),2):
    NEXT J:
    GOTO 14
```

Locate contour line when two nodes of an element have the desired contour value.

```
22  EX(1) = NC(N1,1):
    EY(1) = NC(N1,2):
    EX(2) = NC(N2,1):
    EY(2) = NC(N2,2):
    GOTO 14

24  RETURN
```

(26-32) Clip lines that extend outside the plot bounds.

```
26  C(1) = 1:
    C(2) = 1:
    FOR EN = 1 TO 2:
        D%(EN,1) = (EX(EN) <SL):
        D%(EN,2) = (EX(EN) >SR):
        D%(EN,3) = (EY(EN) <ST):
        D%(EN,4) = (EY(EN) >SB):
    NEXT EN:
    LO = FN AO(I):
    ON LO = 1 GOTO 24

28  FOR EN = 1 TO 2:
        IF D%(EN,1) = 1 OR D%(EN,2) = 1 THEN
            TV = SL * D%(EN,1) + SR * D%(EN,2):
            SP = (EY(2) - EY(1)) / (EX(2) - EX(1)):
            EY(EN) = EY(1) + (TV - EX(1)) * SP:
            EX(EN) = TV:
            D%(EN,1) = 0:
            D%(EN,2) = 0:
            D%(EN,3) = (EY(EN) <ST):
            D%(EN,4) = (EY(EN) >SB):
            LO = FN AO(I):
            ON LO = 1 GOTO 32:
            C(EN) = 2

30      IF D%(EN,3) = 1 OR D%(EN,4) = 1 THEN
            TV = ST * D%(EN,3) + SB * D%(EN,4):
            SP = (EX(2) - EX(1)) / (EY(2) - EY(1)):
            EX(EN) = EX(1) + (TV - EY(1)) * SP:
            EY(EN) = TV:
            D%(EN,1) = (EX(EN) <SL):
            D%(EN,2) = (EX(EN) >SR):
            D%(EN,3) = 0:
            D%(EN,4) = 0:
            LO = FN AO(I):
            C(EN) = 2

32      EN = EN + LO:
    NEXT EN:
    RETURN
```

Plot a solid line.

```
34 HPLOT EX(1),EY(1) TO EX(2),EY(2):
   RETURN
```

Plot a dashed line.

```
36 X1 = EX(1):
   X2 = EX(2):
   Y1 = EY(1):
   Y2 = EY(2):
   LL =  FN LE(I):
   ND = 2 *  INT (LL / 6 +  .5) +  1:
   SX = (X2 - X1) / ND:
   SY = (Y2 - Y1) / ND:
   FOR J = 1 TO ND STEP 2:
       HPLOT X1 +  (J - .8) * SX,Y1 +  (J - .8) * SY TO X1
          +  (J - .2) * SX,Y1 +  (J - .2) * SY:
   NEXT J:
   RETURN
```

(38-40) Determine the centroid of the portion of an element that lies inside the plot bounds.

```
38 XC = 0:
   YC = 0:
   C = 0:
   FOR LI = 1 TO 3:
       I1 = ND%(EL,LI):
       I2 = ND%(EL, FN P2(LI)):
       EX(1) = NC(I1,1):
       EY(1) = NC(I1,2):
       EX(2) = NC(I2,1):
       EY(2) = NC(I2,2):
       GOSUB 26:
       IF LO = 0 THEN
            FOR K = 1 TO 2:
               C = C +  C(K):
               XC = XC +  C(K) * EX(K):
               YC = YC +  C(K) * EY(K):
       NEXT K

40     LI = LI +  3 * (C >4):
   NEXT LI:
   ON C = 0 GOTO 24:
   XC = XC / C:
   YC = YC / C:
   LO =  FN W(A):
   RETURN
```

Determine the method of plotting a line depending on the clipping required.

```
42 ON (ZN = 0) * (ZT = 0) * LT GOTO 34,36:
   ON ZT >0 GOSUB 26:
   ON ((ZT = 0) OR (ZT >0 AND LO = 0))
     * LT GOSUB 34,36:
   RETURN
```

(44-46) Draw a symbol, move it, and select the location of limits for a plot.

```
44 XDRAW 67 AT XP,YP:
   GOSUB 6:
   XDRAW 67 AT XP,YP:
   XP = XP - (AS% = 74) * CT% * (XP >4 +  CT%)
     +  (AS% = 75) * CT% * (XP <272 - CT%):
   YP = YP - (AS% = 73) * CT% * (YP >7 +  CT%)
     +  (AS% = 77) * CT% * (YP <185 - CT%):
   & D(V):
```

```
    IF YP <155 THEN
        & D(G)

46  ON G$ < >"S" AND G$ < >"Q" AND
      G$ < >"<" AND G$ < >">" GOTO 44:
    XDRAW 65 AT XP,YP:
    RETURN
```

(48-50) Move a label and save coordinates for its new location.

```
48  A%(N) = 1:
    GOSUB 6:
    C = (AS% < >81):
    D = (AS% < >88):
    ON C AND D GOSUB 10:
    XP = XP - (AS% = 74) * CT% * (XP >7 + CT%)
       + (AS% = 75) * CT% * (XP <272 - CT%):
    YP = YP - (AS% = 73) * CT% * (YP >7 + CT%)
       + (AS% = 77) * CT% * (YP <185 - CT%):
    I%(N) = I%(N) * D - I%(N) * NOT D:
    ON C GOSUB 10:
    ON C GOTO 48:
    IF CF = 0 THEN
        XL(N) = XP:
        YL(N) = YP

50  A = 0:
    ON CF = 0 GOTO 24:
    CV(N,1) = XP:
    CV(N,2) = YP:
    RETURN
```

Change display to graphics screen with text on bottom four lines.

```
52  VTAB 21:
    POKE 34,20:
    HOME :
    & D(G):
    RETURN
```

Change display to text screen with top two lines outside the window and the rest of the screen erased.

```
54  POKE 34,2:
    HOME :
    & D(T):
    VTAB 4:
    RETURN
```

(56-74) Draw symbols for all user specified boundary conditions.

```
56  L = 15:
    FOR NU = 1 TO NI:
        SK = 0:
        N = BI(NU,0):
        BT = BI(NU,1):
        VB = BI(NU,4):
        ON BT = 1 OR BT = 2 GOTO 58:
        BD = BI(NU,2):
        ON BD = 4 GOTO 64:
        NO = BN%(N,1):
        N0 = BN%(N + 1 - NB * (N = NB),1):
        X1 = X(NO,1):
        Y1 = X(NO,2):
        X2 = X(N0,1):
        Y2 = X(N0,2):
        GOTO 60
```

Draw symbol for nodal temperature or source boundary condition.

```
58     XC = NC(N,1):
       YC = NC(N,2):
       SK =  FN W(A):
       ON SK GOTO 62:
       SCALE= 1:
       FOR K = 0 TO 3:
           ROT= 16 * K:
           DRAW (70 +  (VB <0)) * (BT = 1)
              +  72 * (BT = 2) AT XC,YC:
           K = K +  4 * (BT = 2):
       NEXT K:
       ROT= 0:
    NEXT NU:
    RETURN
```

(60-62) Draw symbol for a convection boundary conditions.

```
60  LS =  FN LE(I):
    NX = (Y2 - Y1) / LS:
    NY = (X1 - X2) / LS:
    ON BT = 3 GOTO 66:
    EX(1) =  FN PX((3 * X1 +  X2) / 4) +  5 * NX:
    EY(1) =  FN PY((3 * Y1 +  Y2) / 4) - 5 * NY:
    EX(2) =  FN PX((3 * X2 +  X1) / 4) +  5 * NX:
    EY(2) =  FN PY((3 * Y2 +  Y1) / 4) - 5 * NY:
    LT = 2

62  ON SK = 0 GOSUB 42:
    LT = 1:
    NEXT NU:
    RETURN
```

Draw symbol for a boundary condition on element face.

```
64  EL = BI(NU,0):
    EX(1) = EC(EL,1) - L / 2:
    EY(1) = EC(EL,1):
    EX(2) = EX(1) +  L:
    EY(2) = EY(1):
    NX = 1:
    NY = 0:
    LT = 2:
    ON BT = 4 GOTO 62:
    XC = EC(EL,1):
    YC = EY(1):
    SK =  FN W(A):
    ON SK GOTO 62:
    DRAW 67 AT XC,YC:
    NEXT NU:
    RETURN
```

Draw symbol for surface heat flux boundary condition.

```
66  NX = ((BD = 1) +  NX * (BD = 3)) *  SGN (VB):
    NY = ((BD = 2) +  NY * (BD = 3)) *  SGN (VB):
    EX(1) =  FN PX((X1 +  X2) / 2):
    EY(1) =  FN PY((Y1 +  Y2) / 2):
    EX(2) = EX(1) +  L * NX:
    EY(2) = EY(1) - L * NY:
    LT = 1:
    GOTO 62
```

(68-74) Draw arrowhead.

```
68  X1 = EX(1):
    Y1 = EY(1):
    X2 = EX(2):
```

```
     Y2 = EY(2):
     ON X2 <10 OR X2 >270 OR Y2 <10 OR
        Y2 >180 GOTO 24:
     LL =  FN LE(I):
     DX = (X2 - X1) / 10:
     DY = (Y2 - Y1) / 10:
     NX =  - DY:
     NY = DX:
     HPLOT X2,Y2 TO X2 - 4 * DX - NX,Y2 - 4
        * DY - NY TO X2 - 3 * DX,Y2 - 3 * DY TO X2 - 4
        * DX +  NX,Y2 - 4 * DY +  NY
     TO X2,Y2:
     RETURN

70   LF = MF * (MF <3) +  3 * (MF >2):
     MG =  ABS (V0) * ( ABS (V0) > ABS (V1))
        +  ABS (V1) * ( ABS (V1) > ABS (V0)):
     FOR I = 1 TO NR:
         I0 = I - I * (I >NN):
         XC = NC(I0,1):
         YC = NC(I0,2):
         IF MO = 6 THEN
             XC = EC(I,1):
             YC = EC(I,2):

72       IF  FN W(A) < >1 THEN
              SCALE= 2:
             DRAW 64 AT XC,YC:
             SCALE= 1:
             AN = RD * AN(I):
             DX = 9 *  ABS (VA(I)) * LF *  COS (AN) / MG:
             DY = 9 *  ABS (VA(I)) * LF *  SIN (AN) / MG:
             EX(1) = XC +  DX:
             EY(1) = YC - DY:
             EX(2) = XC - DX:
             EY(2) = YC +  DY:
             GOSUB 42:
             GOSUB 68

74   NEXT I:
     RETURN
```

(76-80) Fill elements with dot density to indicate value.

```
76   FOR EL = 1 TO NE:
         N% = NE - EL +  1:
         ON ZT >0 GOSUB 142:
         VV = VA(EL):
         IF VV > = V0 AND VV < = V1 THEN
             XC = EC(EL,1):
             YC = EC(EL,2):
             LO =  FN W(A):
             ON LO = 1 GOSUB 38:
             IF LO = 0 THEN
                 KD =  INT ((NI - .0001) * (VV - V0)
                     / (V1 - V0)):
                 C% = CO(KD):
                 O% = OP(KD):
                 PRINT :
                 & F(C%,O%,XC,YC)

78       ON ZT >0 GOSUB 142:
     NEXT EL:
     GOSUB 136:
     POKE 34,2:
```

```
      HOME :
      IF MO = 5 OR MO = 6 THEN
           PRINT "[M]COMPONENT: ";:
          INVERSE :
          PRINT GC$(GC):
          NORMAL

80    PRINT "[M]VALUE RANGES FOR SPARSE TO DENSE DOTS[M]
         OR DARK TO BRIGHT SCREEN:[M]":
      FOR I = 1 TO NI:
          PRINT "  "V0 + (I - 1) * DV; TAB( 18)" TO "V0
             + I * DV:
      NEXT I:
      GOSUB 140:
      & I("[M]TYPE CTRL-V FOR PICTURE,[M]
         ELSE <RETURN>";"2";A$):
      GOSUB 140:
      RETURN
```

Draw a rectangle enclosing the plot.

```
82    HPLOT SL - 1,SB + 1 TO SL - 1,ST - 1 TO SR
         + 1,ST - 1 TO SR + 1,SB + 1 TO SL - 1,SB + 1:
      RETURN
```

(84-88) Determine locations and draw lines for selected contour values.

```
84    & C(CL,CV):
      DIM CL(4,2),CV(NI,2):
      FOR CL = 0 TO NI:
          CV(CL,0) = 99:
          PF = 0:
          N% = NI - CL + 1:
          GOSUB 142:
          CV = V0 + CL * DV:
          FOR EL = 1 TO NE:
              P = 0:
              FOR I = 1 TO DE:
                  N1 = ND%(EL,I):
                  N2 = ND%(EL, FN P2(I)):
                  S1 = VA(N1):
                  S2 = VA(N2):
                  A% = ((S1 - CV) * (CV - S2) > = 0)
                     * (1 + (S1 = S2))

86                ON A% GOSUB 20,22:
          NEXT I,EL:
          GOSUB 142:
      NEXT CL:
      & C(CL):
      & W((CV,0 TO NI,0),0,NI,N%,BR%,ER%):
      P = ER%:
      ON N% = 0 GOTO 24:
      GOSUB 52:
      & I("MARK HI AND LO VALUES","N";A$,"3",0,1):
      ON A$ = "N" GOTO 24:
      XA = 0:
      YA = .1:
      L$ = "LO"

88    XP = CV(0,1):
      YP = CV(0,2):
      ON NOT (XP <SL OR XP >SR OR YP >SB OR
         YP <ST) GOSUB 10:
      L$ = "HI":
      XP = CV(P,1):
```

```
    YP = CV(P,2):
    ON  NOT (XP <SL OR XP >SR OR YP >SB OR
       YP <ST) GOSUB 10:
    RETURN
```

(90-92) Retrieve boundary conditions (RE=10).

```
90  RE = 10:
    GOSUB 224:
    GOSUB 222:
    ON SI%(RE) = 0 GOTO 220:
    NI = N1:
    & C(BI):
    DIM BI(NI,4):
    ONERR  GOTO 240

92  EF = 0:
    GOSUB 228:
    & R(BI,N$):
    ON EF GOTO 92:
    POKE 216,0:
    GOSUB 234:
    GOTO 4
```

(94-96) Retrieve a one-dimensional array and define its range of values.

```
94  GOSUB 224:
    GOSUB 222:
    ON SI%(RE) = 0 GOTO 220:
    & C(VA):
    NR = N1:
    DIM VA(NR):
    ONERR  GOTO 240

96  EF = 0:
    GOSUB 228:
    & R(VA,N$):
    & O(VA,1 TO NR,0,A):
    V0 = VA(1):
    V1 = VA(NR):
    & R(VA,N$):
    ON EF GOTO 96:
    POKE 216,0:
    GOSUB 234:
    GOSUB 4:
    GOSUB 102:
    RETURN
```

(98-100) Retrieve a two-dimensional array.

```
98  GOSUB 224:
    GOSUB 222:
    ON SI%(RE) = 0 GOTO 220:
    NR = N1:
    N% = N2:
    & C(TG):
    DIM TG(NR,N%):
    ONERR  GOTO 240

100 EF = 0:
    GOSUB 228:
    & R(TG,N$):
    ON EF GOTO 100:
    POKE 216,0:
    GOSUB 234:
    GOTO 4
```

(102-104) Display minimum and maximum of array and request lower and upper limits, and increment to be used in plot.

```
102  NI = 10:
     DV = (V1 - V0) / NI:
     ON TP = 2 GOTO 24:
     PRINT "[M]SPECIFY PLOT RANGE AND INCREMENT.[M][M]
        DATA MINIMUM: "V0"[M]DATA MAXIMUM: "V1"[M][M]
        DEFAULT IS FULL RANGE WITH "
        NI"[M]INCREMENTS OF "DV:
     & I("O.K.? (Y/N)";"Y";A$,"4",0,1,V0 <V1):
     ON A$ < >"N" GOTO 24

104  & I("[M]MINIMUM:";V0;V0,"5",BC%,0,V0 <V1):
     ON BC% GOTO 102,24,104:
     & I("MAXIMUM:";V1;V1,"6",BC%,0,V1 >V0):
     ON BC% GOTO 104,24,104:
     & I("INCREMENT:";DV;DV,"7",BC%,0,DV <(V1 - V0)
        AND DV > = (V1 - V0) / 10.0001):
     NI = INT (1.00001 * (V1 - V0) / DV):
     RETURN
```

Plot mesh lines.

```
106  FOR LI = 1 TO NL:
        FOR I = 1 TO 2:
           N% = LN%(LI,I):
           EX(I) = NC(N%,1):
           EY(I) = NC(N%,2):
        NEXT I:
        GOSUB 42:
     NEXT LI:
     RETURN
```

Plot boundary lines.

```
108  A = 21:
     FOR B = 1 TO NS:
        FOR N = IS(2 * B - 1) TO IS(2 * B):
           N1 = N + FN M(B):
           I1 = BN%(N,1):
           I2 = BN%(N1,1):
           EX(1) = NC(I1,1):
           EY(1) = NC(I1,2):
           EX(2) = NC(I2,1):
           EY(2) = NC(I2,2):
           GOSUB 42:
     NEXT N,B:
     RETURN
```

Draw a matrix of square cells representing equal-sized plot zones.

```
110  HCOLOR= 3 * Z%(I,J):
     FOR I1 = 1 TO 3:
        HPLOT FN ZX(I) - 1, FN ZY(J) - 2
           + I1 TO FN ZX(I) + 1, FN ZY(J) - 2 + I1:
     NEXT I1:
     RETURN
```

(112-116) Define the lower-left and upper-right corners of a single-zone plot.

```
112  GOSUB 52:
     PRINT "DEFINE ";:
     INVERSE :
     PRINT "LOWER LEFT CORNER":
     NORMAL :
     PRINT M$"CURSOR[M]"S$"CORNER":
     POKE 34,23:
```

```
     GOSUB 4:
     ON BC% >0 GOTO 24:
     POKE 34,0:
     VTAB 2:
     & D(V):
     XP = SL:
     YP = SB - 10:
     GOSUB 44:
     ON G$ = "<" OR G$ = ">" GOTO 24:
     X1 =  FN UX(XP):
     Y1 =  FN VY(YP):
     XC = XP:
     YC = YP

114  & D(G):
     VTAB 21:
     PRINT "DEFINE ";:
     INVERSE :
     PRINT "UPPER RIGHT CORNER":
     NORMAL :
     VTAB 24:
     CALL  - 868:
     VTAB 2:
     XP =  FN PX(UH):
     YP = ST:
     GOSUB 44:
     ON G$ = "<" GOTO 112:
     ON G$ = ">" GOSUB 10:
     ON G$ = ">" GOTO 48:
     X2 = FN UX(XP):
     Y2 = FN VY(YP):
     XDRAW 65 AT XC,YC

116  XDRAW 65 AT XP,YP:
     GOSUB 132:
     HOME :
     GOSUB 134:
     GOSUB 132:
     ON G$ = "N" GOTO 112:
     UL = X1:
     UH = X2:
     VL = Y1:
     VH = Y2:
     GOSUB 202:
     SR =  FN PX(UH):
     ST =  FN PY(VH):
     RETURN
```

(118-126) Define equal-sized zone plots and select zones to be plotted.

```
118  GOSUB 52:
     & I("MAGNIFICATION?[M] (INTEGER 2 TO 6)";2;MF,"8",
       BC%,1,MF >1 AND MF <7):
     ON BC% GOTO 24,24,118:
     GOSUB 128:
     & D(V):
     GOSUB 134:
     A = (G$ = "N"):
     ON A GOSUB 128:
     ON A GOTO 118:
     A$ = "ZONE PLOTTING OPTIONS:"

120  GOSUB 52:
     PRINT A$"[M]1. PLOT ALL ZONES  3. SELECT ZONES[M]
       2. OMIT ZONES        0. NONE":
```

```
& I(SR$;1;ZS,"9",BC%,1,ZS > = 0 AND ZS <4):
ON BC% < >0 GOSUB 128:
ON BC% GOTO 118,24,120:
& C(Z%):
DIM Z%(MF,MF):
IF ZS <2 THEN
     FOR I = 1 TO MF:
        FOR J = 1 TO MF:
            Z%(I,J) = 1:
NEXT J,I:
GOTO 128

122 GOSUB 52:
PRINT M$"PT IN MATRIXAT TOP RIGHT CORNER, "S$"ZONES ";:
INVERSE :
PRINT ZS$(ZS - 2);:
NORMAL :
PRINT " PLOTTING, <Q>TO QUIT.[M]ZONE ";:
K = MF - 1:
FOR I = 1 TO K:
    J = I - 1:
    HPLOT 250,13 +  4 * J TO 252 +  4 * K,13 +  4 * J:
    HPLOT 253 +  4 * J,10 TO 253 +  4 * J,12 +  4 * K:
NEXT I:
I = 1:
J = 1

124 HCOLOR= 3 - 3 * Z%(I,J):
VTAB 24:
HTAB 6:
PRINT MF * (J - 1) +  I;" ";:
VTAB 2:
HPLOT  FN ZX(I), FN ZY(J):
GET G$:
A =  ASC (G$):
HCOLOR= 3 * Z%(I,J):
HPLOT  FN ZX(I), FN ZY(J):
I = I +   FN WM(I) * (G$ = "J" OR G$ = "K"):
J = J +   FN WM(J) * (G$ = "I" OR G$ = "M"):
IF G$ = "S" THEN
    Z%(I,J) =  NOT Z%(I,J):
    GOSUB 110

126 ON G$ < >"Q" GOTO 124:
ON ZS = 3 GOTO 24:
FOR I = 1 TO MF:
    FOR J = 1 TO MF:
        Z%(I,J) =  NOT Z%(I,J):
NEXT J,I:
RETURN
```

(128-130) Label corners of indicated equal-sized zone plots with the zone numbers.

```
128 ROT= 16:
FOR I = 1 TO MF - 1:
    PB = SB - I * (SB - ST) / MF:
    FOR J = SL TO SR STEP 15:
        XDRAW 69 AT J,PB:
NEXT J,I:
ROT= 0:
FOR J = 1 TO MF - 1:
    PL = SL +  J * (SR - SL) / MF:
    FOR I = SB TO ST STEP  - 15:
        XDRAW 69 AT PL,I:
NEXT I,J
```

```
130  ZN = 0:
     FOR I = 1 TO MF:
         PB = SB - (I - 1) * (SB - ST) / MF:
         PT = PB - (SB - ST) / MF:
         FOR J = 1 TO MF:
             PL = SL +  (J - 1) * (SR - SL) / MF:
             PR = PL +  (SR - SL) / MF:
             ZN = ZN +  1:
             L$ =  STR$ (ZN):
             XP = PL:
             YP = PT:
             XA = 1.5:
             YA =  - 1.5:
             GOSUB 10:
             XP = PR:
             YP = PB:
             XA =  - 1.5:
             YA = 1.5:
             GOSUB 10:
     NEXT J,I:
     RETURN
```

Use lines of symbols to illustrate division of a plot into equal-sized plots.

```
132  PL =  FN PX(X1):
     PR =  FN PX(X2):
     PB =  FN PY(Y1):
     PT =  FN PY(Y2):
     FOR I = PT +  5 TO PB STEP 15:
         XDRAW 69 AT PL,I:
         XDRAW 69 AT PR,I:
     NEXT I:
     ROT= 16:
     FOR I = PL +  5 TO PR STEP 15:
         XDRAW 69 AT I,PT:
         XDRAW 69 AT I,PB:
     NEXT I:
     ROT= 0:
     RETURN
```

Ask user to approve the zone selection.

```
134  L$ = "O.K.? (Y/N)":
     XP = 15:
     YP = 191:
     XA = 1:
     YA = 1:
     & D(V):
     GOSUB 10:
     GOSUB 6:
     GOSUB 10:
     ON AS% < >78 AND AS% < >89 GOTO 134:
     RETURN
```

(136-142) Display a message or status indicator on the graphics screen.

```
136  L$ = "DRAWING SINGLE ZONE":
     ON ZT = 1 GOTO 138:
     ON ZN = 0 GOTO 24:
     L$ = "DRAWING ZONE " +  STR$ (ZN) +  "..."

138  XP = 52:
     YP = 0:
     XA = 1:
     YA =  - 1:
     GOSUB 10:
     L$ = "PLEASE BE PATIENT":
```

```
       XP = 72:
       YP = 191:
       YA = 1:
       GOTO 10

140  L$ = "TYPE CTRL-T FOR LEGEND, ELSE <RETURN>":
       XP = 1:
       YP = 191:
       XA = 1:
       YA = 1:
       GOTO 10

142  L$ =  STR$ (N%):
       XP = 260:
       YP = 91:
       XA = 1:
       YA = 1:
       GOTO 10
```

(144-158) Plot the entire mesh, offer zone plotting options, and plot the selected zones.

```
144  GOSUB 54:
       PRINT "[M]PLOTS:[M]  WHOLE BODY[M]  ZONES (ZOOM)[M][M]
          LABELS:[M]  ELEMENTS[M]  NODES[M]  CONTOURS[M]
          NONE[M]  (USE REDRAW TO ENHANCE LABEL CLARITY)[M]":
       GOSUB 4:
       ON BC% >0 GOTO 24

146  ZN = 0:
       SR = RS:
       ST = TS:
       SC = S0:
       UL = XL:
       UH = XH:
       VL = YL:
       VH = YH:
       MF = 1:
       ON ZT = 1 GOSUB 202:
       ZT = 0:
       GOSUB 160:
       ON BC% = 2 GOTO 158:
       GOSUB 54:
       HOME :
       PRINT "  WHOLE-BODY PLOT IS COMPLETED.
          PLOTS OFSELECTED  PARTS  OF THE  BODY MAY NOW BEMADE.
          OPTIONS ARE AVAILABLE FOR EITHER A";

148  PRINT "SINGLE ZONE OF ANY SIZE OR  FROM 1 TO
          36EQUAL-SIZED  ZONES. ANY OF THESE  EQUAL-SIZED  ZONES
          MAY BE DEVELOPED INTO A NEWPLOT WHICH CAN
          BE PRINTED  AND  COMBINEDWITH THE OTHERS TO
          PRODUCE  AN  ENLARGEDPLOT OF THE BODY.":
       GOSUB 4

150  GOSUB 52:
       HOME :
       & I("ZONE PLOTTING OPTIONS:[M]
          1. SINGLE ZONE      0. NONE[M]
          2. MULTIPLE EQUAL-SIZED ZONES",1;ZT,"10",BC%,
            1,ZT > = 0 AND ZT <3):
       ON BC% GOTO 24,24,150

152  ON ZT = 0 GOTO 24:
       ON ZT GOSUB 112,118:
```

```
        ON BC% = 1 GOTO 150:
        ON ZT = 1 AND BC% = 2 GOTO 24:
        MF = MF * (ZT = 2) +  (ZT = 1) * 2:
        ON ZT = 1 GOSUB 160:
        ON ZT = 1 AND BC% = 2 GOTO 24:
        ON ZT = 1 GOTO 158:
        SC = S0 * MF:
        FOR ZJ = 1 TO MF:
        FOR ZI = 1 TO MF:
             ON Z%(ZI,ZJ) = 0 GOTO 156


154     ZN = (ZJ - 1) * MF +  ZI:
         V% = 1 +  (ZN - 1) / MF:
         H% = ZN - MF * (V% - 1):
            FOR ZI =) / MF:
                 H% = ZN - MF * (V% - 1):
                 VL = YL +  (V% - 1) * (SB - ST) / SC:
                 VH = YL +  V% * (SB - ST) / SC:
                 UL = XL +  (H% - 1) * (SR - SL) / SC:
                 UH = XL +  H% * (SR - SL) / SC:
                 GOSUB 160:
                 ON BC% = 2 GOTO 156:
                 GOSUB 52:
                 & I("REPEAT LAST ZONE";"N";A$,"11",0,1):
                 ON A$ = "Y" GOTO 154

156              ZI = ZI +  MF * (BC% = 2):
           NEXT ZI:
           ZJ = ZJ +  MF * (BC% = 2):
      NEXT ZJ

158   GOSUB 52:
      ON ZT = 0 GOTO 24:
      & I("[M]MORE ZONE PLOTS? (Y/N)";"N";A$,"12",0,1):
      ON A$ = "Y" GOTO 144:
      RETURN
```

(160-196) Produce a complete plot using shading, arrows or contours. Label elements/nodes/contour lines, add extra labels, move/redraw, provide plots, and save.

```
160   GOSUB 328:
      GOSUB 136:
      LT = (GD = 1) +  2 * (GD = 3):
      ON GD = 1 OR GD = 3 GOSUB 106:
      LT = 1:
      ON GD = 2 OR GD = 3 GOSUB 108:
      ON TP = 2 GOSUB 70:
      ON TP = 3 GOSUB 84:
      IF TP = 1 THEN
           HCOLOR= 3:
         ON MF >1 GOSUB 82:
         GOSUB 76:
         HCOLOR= 0:
         ON MF >1 GOSUB 82:
         HCOLOR= 3

162   ON MO = 2 GOSUB 56:
      ON TP < >1 GOSUB 136:
      GOSUB 52:
      & I("LABEL PLOT (ANY TYPE)? (Y/N)";"N";A$,"13",0,1):
      ON A$ < >"Y" GOTO 194:
      LA = 1
```

(164-178) Label elements on nodes and edit.

```
164 GOSUB 52:
    & I("LABEL " + PV$(LA) + "S";"N";A$,"14",0,1):
    ON A$ < >"Y" GOTO 178:
    CF = 0:
    GOSUB 54:
    PRINT "LABEL OPTIONS:[M][M]
       1. ALL "PV$(LA)"S (DEFAULT POSITIONS)[M]
       2. SELECTED "PV$(LA)"S (USER POSITIONED)[M][M]
       0. NONE[M]":
    N = 1

166 & I(SR$,1;LN,"15",BC%,1,LN > = 0 AND LN <3):
    ON BC% < >0 OR LN = 0 GOTO 180:
    N = NE * (LA = 1) + NN * (LA = 2):
    & C(A%,I%,XL,YL):
    DIM A%(N),I%(N),XL(N),YL(N):
    ON LA GOSUB 322,326:
    N = 1:
    ON LN = 2 GOTO 172:
    ON NO = 0 GOTO 170:
    & D(V):
    XA = (LA = 2):
    YA = 0

168 FOR I = 1 TO NO:
       L$ = LA$(LA) + STR$ (I%(I)):
       XP = XL(I):
       YP = YL(I):
       GOSUB 10:
       A%(I) = 1:
    NEXT I:
    GOTO 176

170 HOME :
    VTAB 12:
    INVERSE :
    PRINT "NO "PV$(LA)"S FALL WITHIN THIS ZONE.":
    NORMAL :
    GOSUB 4:
    GOTO 178

172 GOSUB 54:
    PRINT "INSTRUCTIONS:[M][M]USE ARROWS TO MOVE CURSOR.[M]
       USE "S$"THE LABEL TO BE[M] CHANGED.[M]
       USE <X>TO ADD/REMOVE A LABEL.[M]"
       M$"LABEL IN[M] LARGE INCREMENTS.[M]
       USE <CONTROL>WITH THESE LETTERS TO[M]
       MOVE IN SMALL INCREMENTS."

174 PRINT "USE <Q>TO QUIT LABEL EDIT.":
    & I(WR$;"";A$,"16"):
    & D(V):
    N = N * (N >0 AND N < = NO)
       + (N <1 OR N >NO):
    XA = (LA = 2):
    YA = 0:
    GOSUB 12

176 GOSUB 52:
    & I("EDIT " + PV$(LA) + " LABELS";"N";A$,"17",0,1):
    ON A$ = "Y" GOTO 172:
    HOME :
```

```
     & I("REDRAW " +  PV$(LA) +  " LABELS";"N";A$,"18",0,1):
     ONA$ <>"Y" GOTO 178:
     & D(V):
     FOR I = 1 TO NO:
         L$ = LA$(LA) +   STR$ (I%(I)):
         XP = XL(I):
         YP = YL(I):
         ON I%(I) >0 AND A%(I) GOSUB 8:
     NEXT I

178  & D(G):
     HOME :
     LA = LA +  1:
     & C(A%,I%,XL,YL):
     ON LA <3 GOTO 164:
     LA = 1
```

(180-186) Label contour lines.

```
180  CF = 1:
     ON TP < >3 GOTO 188:
     GOSUB 52:
     & I("LABEL CONTOUR LINES";"N";A$,"19",BC%,1):
     ON BC% GOTO 164,24,180:
     ON A$ < >"Y" GOTO 188:
     HOME :
     PRINT "OPTIONS:[M]1. USE LETTER" TAB( 20)"0. NONE[M]
        2. USE VALUE":
     & I(SR$;1;CO,"20",BC%,1,CO = 0 OR CO = 1 OR CO = 2)

182  ON BC% GOTO 180,188,180:
     HOME :
     PRINT "ARROWS MOVE CURSOR[M]<S>SELECTS CONTOUR;
        <X>ADDS/DELETES    <I><J>
        <K><M>MOVE LABEL; <Q>QUITS":
     & I(WR$;"";A$,"21"):
     & D(V):
     XA = 0:
     YA = 0:
     & C(A%,I%):
     DIM A%(P),I%(P):
     FOR I = 0 TO P:
        I%(I) =  - 1:
     NEXT I:
     N = 0:
     GOSUB 16

184  IF CO = 1 THEN
         POKE 34,2:
        HOME :
        INVERSE :
        PRINT GC$(GC * (MO = 5)):
        NORMAL :
        PRINT "[M]LETTER      VALUE[M]":
        FOR I = 0 TO NI:
           PRINT "   " CHR$ (I +  65) TAB( 12)V0 +  I * DV:
        NEXT I:
        GOSUB 140:
        VTAB 21:
        & I("TYPE CTRL-V TO VIEW, ELSE <RETURN>";
           "";A$,"22"):
        & D(V):
        GOSUB 140
```

```
186  GOSUB 52:
     & I("'REDRAW' CONTOUR LABELS? (Y/N)";"N";A$,"23",0,1):
     ON A$ < >"Y" GOTO 188:
     XA = 0:
     YA = 0:
     & D(V):
     FOR N = 0 TO P:
        CL = CV(N,0):
        CV = V0 +  CL * DV:
        L$(1) =  CHR$ (CL +  65):
        L$(2) =  STR$ (CV):
        L$ = L$(CO):
        XP = CV(N,1):
        YP = CV(N,2):
        ON I%(N) >0 GOSUB 8:
     NEXT N:
     & C(A%,I%):
     & D(G)
```

(188-192) Add other labels to the plot.

```
188  GOSUB 52:
     & I("OTHER LABELS? (Y/N)";"N";A$,"24",BC%,1):
     ON BC% GOTO 164,194,188:
     ON A$ < >"Y" GOTO 194:
     HOME :
     & I("LABEL:";"";L$,"25",BC%,30):
     ON BC% GOTO 188,194,188

190  HOME :
     PRINT "LOCATE CENTER OF LABEL[M]"M$"CURSOR[M]"S$"POINT":
     VTAB 2:
     XA = 0:
     YA = 0:
     XP = 140:
     YP = 12:
     GOSUB 44:
     XDRAW 65 AT XP,YP:
     GOSUB 10:
     GOSUB 52:
     & I("EDIT LABEL? (Y/N)";"N";A$,"26",0,1):
     ON A$ < >"Y" GOTO 192:
     & D(V):
     N = 1:
     GOSUB 48

192  GOSUB 52:
     & I("'REDRAW' LABEL? (Y/N)";"N";A$,"27",0,1):
     ON A$ < >"Y" GOTO 188:
     GOSUB 8:
     GOTO 188
```

Draw '+' symbols at corners of multiple-zone plots.

```
194  IF ZT = 2 THEN
         & D(V):
        SCALE= 1:
        DRAW 64 AT SL,ST:
        DRAW 64 AT SL,SB:
        DRAW 64 AT SR,ST:
        DRAW 64 AT SR,SB:
        GOSUB 52:
        PRINT "ZONE "ZN" IS COMPLETE.[M]
           TYPE CTRL-V FOR VIEWING FULL SCREEN":
        GOSUB 4
```

Specify disposition of generated plot.

```
196  GOSUB 54:
     PRINT "PLOT DISPOSITION OPTIONS:[M][M]
        1. PROCEED WITH ZONE(S)[M]
        2. SAVE PLOT TO DISKETTE[M][M]
        0. END THIS PLOT[M]
     ":
     & I(SR$,0;A,"28",BC%,1,A > = 0 AND A <3):
     ON BC% < >0 GOTO 196:
     BC% = 2 * (A = 0):
     ON A <2 GOTO 24
```

(198-200)Save plot to a data file.

```
198  HOME :
     PRINT "SAVE PICTURE[M][M]
        INSERT DATA DISKETTE, IF NECESSARY,[M]
        INTO DRIVE "DR"[M]":
     GOSUB 4:
     ON BC% GOTO 196,196,198:
     A$ = MID$ (MU$(MO),6,17):
     & I("[M]FILE NAME?[M]('" + KW$ + "/'"
        + " PREFIX AND[M]'.PIC' SUFFIX WILL BE ADDED)",
        A$,A$,"29",BC%,17)

200  A$ = KW$ + "/" + A$ + ".PIC":
     PRINT D$"BSAVE "A$",A$2000,L$1FF8":
     PRINT "[M]PLOT SAVED IN FILE:":
     INVERSE :
     PRINT A$:
     NORMAL :
     GOSUB 4:
     GOTO 196
```

Define plot scale.

```
202  A = (SB - ST) / (VH - VL):
     SC = (SR - SL) / (UH - UL):
     SC = SC * (SC < = A) + A * (SC >A):
     RETURN
```

(206-218) Load element numbers (RE=5), nodal coordinates (RE=6), nodes for lines (RE=7), and define the range of the coordinates.

```
206  RE = 5:
     GOSUB 224:
     GOSUB 222:
     ON SI%(RE) = 0 GOTO 220:
     ONERR GOTO 240

208  EF = 0:
     GOSUB 228:
     & R(ND%,N$):
     ON EF GOTO 208:
     GOSUB 234:
     RE = 6:
     GOSUB 222:
     ON SI%(RE) = 0 GOTO 220:
     & C(X,NC,CC,EC):
     DIM X(NN,2),NC(NN,2),CC(NE,2),EC(NE,2)

210  EF = 0:
     GOSUB 228:
     & R(X,N$):
     ON EF GOTO 210:
     GOSUB 234:
```

```
      FOR I = 1 TO NN:
           X(I,0) = I:
           NC(I,0) = I:
      NEXT I:
      & O(X,1 TO NN,1,A):
      XL = X(1,1):
      XH = X(NN,1):
      & O(X,1 TO NN,2,A):
      YL = X(1,2):
      YH = X(NN,2):
      & O(X,1 TO NN,0,A)

212   RE = 7:
      GOSUB 222:
      ON SI%(RE) = 0 GOTO 220

214   EF = 0:
      GOSUB 228:
      & R(LN%,N$):
      ON EF GOTO 214:
      GOSUB 234:
      RE = 8:
      GOSUB 222:
      ON SI%(RE) = 0 GOTO 220

216   EF = 0:
      GOSUB 228:
      & R(BN%,N$):
      ON EF GOTO 216:
      GOSUB 234:
      PRINT "[M][M]CALCULATING ELEMENT CENTROIDS...";:
```

Calculate element centroids.

```
218   & C(CC,EC):
      DIM CC(NE,2),EC(NE,2):
      FOR EL = 1 TO NE:
           PRINT ".";:
           XC = 0:
           YC = 0:
           FOR I = 1 TO DE:
                N% = ND%(EL,I):
                XC = XC +  X(N%,1):
                YC = YC +  X(N%,2):
           NEXT I:
           CC(EL,1) = XC:
           CC(EL,2) = YC:
           CC(EL,0) = EL:
           EC(EL,0) = EL:
      NEXT EL:
      PRINT :
      & M(CC = CC / (DE)):
      RETURN
```

Error message.

```
220   FLASH :
      ON  PEEK (222) = 77 GOTO 252:
      PRINT "[M][G][G]DATA NOT DEFINED":
      NORMAL :
      PRINT "EITHER THE FILE":
      INVERSE :
      PRINT NA$:
      NORMAL :
      PRINT "DOESN'T EXIST OR THE DATA IS NOT VALID":
```

```
    GOSUB 4:
    BC% = 2:
    RETURN
```

Read the data file descriptors.

```
222 N = FRE (0):
    PRINT D$:
    PRINT D$"OPEN"FI$",L100"DR$(DR):
    PRINT D$"READ"FI$",R"RE:
    INPUT SI%(RE),NA$,DE$,N1,D1$,N2,D2$:
    PRINT D$"CLOSE"FI$:
    GOTO 230
```

(224-226) Verify the existence of a data file.

```
224 ONERR GOTO 248

226 E2 = 0:
    PRINT D$:
    ON E2 GOTO 226:
    POKE 216,0:
    RETURN
```

(228-232) Get the file name for the data to be retrieved.

```
228 ON Z = 2 GOTO 230:
    PRINT "[M]*** ";:
    INVERSE :
    PRINT "LOADING";:
    NORMAL :
    ON Z GOTO 232:
    PRINT " DATA FROM DISKETTE ***[M][M]
       FILE NAME FOR[M]"DE$:
    & I("";NA$,NA$,"30",BC%,30):
    ON BC% < >0 GOTO 228:
    ON LEFT$ (NA$, LEN (KW$)) = KW$ GOTO 230:
    NA$ = KW$ + "/" + NA$

230 N$ = NA$ + DR$(DR):
    RETURN

232 PRINT " ";NA$:
    GOTO 230
```

Print the description of the loaded data.

```
234 PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2" "D2$"[M]
       WERE LOADED":
    RETURN
```

Read the problem description.

```
236 PRINT D$"OPEN"FI$",L100"DR$(DR):
    PRINT D$"READ"FI$",R0":
    INPUT TE,KW$,PD$:
    PRINT D$"CLOSE"FI$:
    RETURN
```

Print the problem description.

```
238 ON LEN (KW$) = 0 GOSUB 236:
    PRINT "[M]PROBLEM KEYWORD: "KW$"[M][M]
       PROBLEM DESCRIPTION:[M]"PD$:
    RETURN
```

Trap errors.

```
240 EF = 1:
    ER = PEEK (222):
    POKE 216,0:
```

```
      ON ER = 255 GOTO 244:
      FLASH :
      ON ER = 77 GOTO 252:
      PRINT "[G][G][M]ERROR ";ER;"[M]":
      NORMAL :
      ER = ER * (ER >3 AND ER <11)
         +    NOT (ER >3 AND ER <11):
      ON ER = 1 OR ER = 5 GOTO 242:
      ON ER >3 AND ER <11 GOTO 246

242   ON MO = 7 GOSUB 318:
      ON MO < >7 GOSUB 222:
      RESUME

244   END

246   POKE 216,0:
      PRINT ER$(ER - 4):
      ON ER = 6 GOSUB 250:
      ON ER = 4 OR ER = 5 OR ER = 8 OR ER = 9 GOSUB 4:
      ON ER <10 GOTO 242:
      & I("UNLOCK? (Y/N)";"Y";A$,"31",0,1):
      ON A$ = "N" GOTO 24:
      PRINT D$"UNLOCK "NA$DR$(DR):
      GOTO 242

248   E2 = 1:
      POKE 216,0:
      ON  PEEK (222) = 255 GOTO 244:
      ON  PEEK (222) = 77 GOTO 252:
      PRINT "[G][G][M]DATA DISKETTE IS NOT IN DRIVE":
      GOSUB 4:
      RESUME

250   & I("[M]CATALOG? (Y/N)";"N";A$,"32",0,1):
      ON A$ = "N" GOTO 24:
      PRINT D$"CATALOG"DR$(DR):
      RETURN

252   PRINT "[G][G][M]SORRY, OUT OF MEMORY":
      NORMAL :
      END
```

(1,254-280) Cold start initialization.

```
254   & D(T):
      Z =  PEEK (2163):
      HOME :
      INVERSE :
      FOR I = 1 TO 4:
         PRINT  TAB( 2)" " TAB( 38)" ":
      NEXT I:
      NORMAL :
      VTAB 2:
      HTAB 3:
      PRINT " APPLIED FINITE ELEMENT ANALYSIS: ":
      HTAB 3:
      PRINT "     AN APPLE " CHR$ (221) CHR$ (219)
         " IMPLEMENTATION     [M][M]":
      HTAB 12:
      PRINT "COPYRIGHT, 1985"

256   VTAB 8:
      HTAB 14:
      INVERSE :
```

```
       PRINT "PLOT";:
       NORMAL :
       PRINT "   (6.0)[M][M]ABSTRACT:[M][M]
          PREPARE FINISHED PLOTS OF GRID, BOUNDARYCONDITIONS,
          OR RESULTS.  ANY PLOT MAY BELABELED AND PRINTED.
          ZONE PLOTS  MAY BESELECTED TO ENLARGE PARTS OF
          INTEREST.":
       GOSUB 4

258    ON NOT ( PEEK (6462 ) = 75 AND PEEK (6463)
          = 75) GOSUB 332:
       & C(DR$):
       DIM DR$(2):
       DR$(0) = " ":
       DR$(1) = ",D1":
       DR$(2) = ",D2":
       DR =  PEEK (2048):
       ON DR = 2 GOTO 262

260    & B(200,20):
       PRINT "INSERT ";:
       FLASH :
       PRINT "DATA";:
       NORMAL :
       PRINT " DISKETTE INTO DRIVE":
       GOSUB 4

262    HOME :
       GOSUB 224:
       HOME :
       GOSUB 238:
       PRINT U$:
       POKE 34,7:
       HOME :
       DD = 2:
       DIM DI$(DD),DF$(4),SI%(22),CO(9),OP(9),ZS$(1),
          MU$(7),PV$(2),TP$(3),GC$(3),LA$(2),L$(2),ER$(6),
          EX(2),EY(2)

264    ER$(0) = "DISKETTE IS WRITE PROTECTED":
       ER$(2) = "FILE NOT FOUND":
       ER$(3) = "VOLUME MISMATCH":
       ER$(4) = "I/O ERROR -[M] DOOR OPEN OR DISK
          NOT INITIALIZED":
       ER$(5) = "DISK IS FULL":
       ER$(6) = "FILE IS LOCKED"

266    RESTORE :
       DATA 20,0,10,1,19,0,5,1,16,1,8,1,8,0,1,1,1,0,0,0:
       FOR K = 0 TO 9:
          READ CO(K),OP(K):
       NEXT K:
       LA$(1) = "E":
       RD =  ATN (1) / 45:
       DF$(1) = "X":
       DF$(2) = "Y":
       DF$(3) = "R":
       DF$(4) = "Z":
       DI$(1) = DF$(2 * TE - 1)

268    DI$(2) = DF$(2 * TE):
       ZS$(0) = "OMITTED FROM":
       ZS$(1) = "INCLUDED IN":
       DEF  FN C(PN) = (A = 21) * (1 - (PN +  1)
```

```
    * (N = PN)) + (A = 8) * ( - 1 + (PN + 1)
    * (N = 0)):
  DEF FN LE(I) = SQR ((X1 - X2) ^ 2
    + (Y1 - Y2) ^ 2):
  DEF FN AO(I) = ((D%(1,1) * D%(2,1) + D%(1,2)
    * D%(2,2) + D%(1,3) * D%(2,3) + D%(1,4)
    * D%(2,4)) >0)

270 DEF FN S(NE) = (A = 21) * (1 - NE * (N = NE))
    + (A = 8) * ( - 1 + NE * (N = 1)):
  DEF FN M(B) = ((N <IS(2 * B)) - (IS(2 * B)
    - IS(2 * B - 1)) * (N = IS(2 * B))) * (A = 21)
    - ((N >IS(2 * B - 1)) - (IS(2 * B)
    - IS(2 * B - 1)) * (N = IS(2 * B - 1))) * (A = 8):
  DEF FN P2(I) = I + 1 - DE * (I = DE):
  DEF FN W(A) = (XC <SL OR XC >SR OR
    YC >SB OR YC <ST)
272 RE = 5:
  GOSUB 222:
  NE = N1:
  DE = N2:
  RE = 6:
  GOSUB 222:
  NN = N1:
  RE = 7:
  GOSUB 222:
  NL = N1:
  RE = 8:
  GOSUB 222:
  NB = N1:
  DIM ND%(NE,DE),LN%(NL,3),BN%(NB,2),XP(DE),YP(DE):
  GOSUB 206:
  ON BC% = 2 GOTO 244:
  NS = BN%(0,0):
  & C(IS):
  DIM IS(2 * NS)

274 FOR I = 1 TO 2 * NS:
      IS(I) = BN%(I,0):
  NEXT I:
  SL = 35:
  SR = 245:
  SB = 175:
  ST = 15:
  RS = SR:
  TS = ST:
  DEF FN PX(UX) = SL + SC * (UX - UL):
  DEF FN PY(VY) = SB - SC * (VY - VL):
  DEF FN UX(PX) = UL + (PX - SL) / SC:
  DEF FN VY(PY) = VL - (PY - SB) / SC:
  UL = XL:
  UH = XH:
  VL = YL:
  VH = YH:
  GOSUB 218:
  GOSUB 202:
  S0 = SC

276 MU$(0) = "EXIT":
  MU$(1) = "PLOT GENERATED GRID":
  MU$(2) = "PLOT BOUNDARY CONDITIONS":
  MU$(3) = "PLOT NODE TEMPERATURES":
  MU$(4) = "PLOT ELEMENT TEMPERATURES":
  MU$(5) = "PLOT NODE TEMP GRADIENTS":
```

```
    MU$(6) = "PLOT ELEMENT TEMP GRADIENTS"

278 MU$(7) = "RETRIEVE PICTURE FROM DISK":
    PV$(1) = "ELEMENT":
    PV$(2) = "NODE":
    TP$(1) = "SHADE ELEMENTS TO SHOW MAGNITUDE":
    TP$(2) = "ARROWS FOR DIRECTION & MAGNITUDE":
    TP$(3) = "CONTOUR LINES":
    GC$(1) = DI$(1) + "-COMPONENT":
    GC$(2) = DI$(2) + "-COMPONENT"

280 GC$(3) = "RESULTANT VECTOR":
    DEF FN WM(I) = - (A = 74 OR A = 77) * ((1 - MF)
       * (I = 1) + (I < >1))
       + (A = 75 OR A = 73) * ((1 - MF) * (I = MF)
       + (I < >MF)):
    DEF FN ZX(I) = 251 + 4 * (I - 1):
    DEF FN ZY(J) = 11 + 4 * (MF - 1) - 4 * (J - 1)
```

(2,282-284) Warm restart and menu.

```
282 POKE 216,0:
    NM = PEEK (2166):
    TEXT :
    & D(T):
    HOME :
    TP = 0:
    PRINT "       PLOT  (6.05)[M][M]
       -- PLOT FORMULATION --[M]
       1. "MU$(1)"[M]2. "MU$(2)"[M][M]
       -- PLOT RESULTS --":
    FOR I = 3 TO 6:
        PRINT I". "MU$(I):
    NEXT I:
    PRINT "[M] -- OTHER --[M]
       7. "MU$(7)

284 PRINT "[M]0. NONE[M]":
    & C(TG,VA):
    & I(SR$,NM;MO,"33",BC%,1,MO > = 0 AND MO <8):
    ON BC% GOTO 1,286,282:
    HOME :
    ON MO = 0 GOTO 286:
    HE$ = MU$(MO) + "  (6." + STR$ (MO) + ")":
    PRINT "**" TAB( (39 - LEN (HE$)) / 2 + 2)
       HE$ TAB( 39)"**[M]":
    POKE 34,2:
    ON (MO + 1) / 2 GOTO 292,306,294,308
```

(286-290) Exit menu.

```
286 HOME :
    PRINT "       PLOT: EXIT  (6.8)[M][M]
       1. EXIT TO MAIN MENU[M]
       2. DON'T EXIT; REMAIN IN 'PLOT'[M][M]
       0. STOP[M]":
    & I(SR$,0;EO,"34",BC%,1,EO > = 0 AND EO <3):
    ON EO = 0 OR BC% = 2 GOTO 244:
    ON EO = 2 GOTO 282

288 POKE 2166,1:
    IF DR = 1 THEN
        & B(200,20):
       PRINT "INSERT ";:
       FLASH :
       PRINT "PROGRAM";:
```

```
          NORMAL :
          PRINT " DISKETTE CONTAINING[M]'";:
          INVERSE :
          PRINT "HELLO";:
          NORMAL :
          PRINT "' INTO DRIVE":
          GOSUB 4:
          ON BC% = 1 OR BC% = 3 GOTO 286

290  PRINT D$"RUN HELLO,D1"
```

Menu options 1 and 2: Plot grid or boundary conditions.

```
292  ON MO = 2 GOSUB 90:
     ON BC% = 2 GOTO 282:
     GD = (MO = 1) + 2 * (MO = 2):
     GOSUB 144:
     & D(T):
     & C(BI):
     SI%(RE) = 0:
     POKE 2166,MO + 1:
     GOTO 282
```

(294-304) Menu options 5 and 6: Plot nodal temperature gradients or element temperature gradients.

```
294  POKE 34,2:
     HOME :
     PRINT "GRADIENT COMPONENT OPTIONS:[M]":
     FOR I = 1 TO 3:
          PRINT I". "GC$(I):
     NEXT I:
     PRINT "[M]0. NONE[M]":
     & I(SR$,1;GC,"35",BC%,1,GC > = 0 AND GC <4):
     IF GC = 0 OR BC% < >0 THEN
        & C(TG,VA):
          SI%(RE) = 0:
          POKE 2166,MO + 1:
          GOTO 282

296  POKE 34,2:
     HOME :
     PRINT "GRADIENT: ";:
     INVERSE :
     PRINT GC$(GC)"[M]":
     NORMAL :
     POKE 34,4:
     TP = (MO = 6 AND GC <3) + 3 * (MO = 5 AND GC <3):
     GD = (TP = 1):
     ON TP >0 GOTO 300

298  HOME :
     PRINT "PLOT TYPES - OPTIONS:[M]":
     FOR I = 1 TO 2:
          PRINT I". "TP$(I + (MO = 5)):
     NEXT I:
     PRINT "[M]0. NONE[M]":
     & I(SR$,1;TP,"36",BC%,1,TP > = 0 AND TP <3):
     ON BC% < >0 OR TP = 0 GOTO 294:
     TP = TP + (MO = 5):
     GD = (TP = 1)

300  HOME :
     PRINT "PLOT: ";:
     INVERSE :
     PRINT TP$(TP)"[M]":
```

```
     NORMAL :
     POKE 34, PEEK (37):
     ON TP < >1 GOSUB 302:
     GOTO 304

302  HOME :
     PRINT "[M][M]GRID/BOUNDARY OPTIONS:[M][M]
        1. PLOT "TP$(TP) SPC( 1 + (TP = 3) * 20)
        "WITH SOLID BOUNDARY LINE[M]
        2. PLOT "TP$(TP) SPC( 1 + (TP = 3) * 20)
        "WITH SOLID BOUNDARY[M]  AND DASHED LINES FOR ELEMENTS[M]":
     & I(SR$,1;GD,"37",BC%,1,GD >0 AND GD <3):
     GD = GD + 1:
     RETURN

304  HOME :
     RE = 19 + (MO = 5):
     GOSUB 98:
     ON BC% = 2 GOTO 282:
     & C(VA,AN):
     DIM VA(NR,AN(NR)):
     FOR I = 1 TO NR:
        VA(I) = TG(I,GC):
     NEXT I:
     & O(VA,1 TO NR,0,A):
     V0 = VA(1):
     V1 = VA(NR):
     FOR I = 0 TO NR:
        VA(I) = TG(I,GC):
        AN(I) = TG(I,DD+ 2):
     NEXT I:
     & C(TG):
     GOSUB 102:
     GOSUB 144:
     & D(T):
     & C(AN,VA):
     GOTO 294
```

Menu options 3 and 4: Plot model temperatures or element temperatures.

```
306  POKE 34,2:
     HOME :
     TP = (MO = 4) + 3 * (MO = 3):
     GD = (MO = 4):
     RE = 17 + (MO = 4):
     GOSUB 94:
     ON BC% < >0 GOTO 282:
     ON MO = 3 GOSUB 302:
     ON BC% < >0 OR GD = 0 GOTO 306:
     GOSUB 144:
     & D(T):
     & C(VA):
     POKE 2166,MO + 1:
     GOTO 282
```

(308-320) Menu option 7: Retrieve picture from disk.

```
308  POKE 34,2:
     HOME :
     & C(PF$):
     DIM PF$(20):
     ON DR = 2 GOTO 310:
     & B(200,20):
     PRINT "[M][M]INSERT ";:
     FLASH :
     PRINT "PROGRAM";:
```

```
     NORMAL :
     PRINT " DISKETTE":
     GOSUB 4:
     ON BC% GOTO 282

310  PRINT D$"BLOAD N,A$1420,D1":
     ON DR = 2 GOTO 312:
     PRINT "[M][M]INSERT ";:
     FLASH :
     PRINT "DATA";:
     NORMAL :
     PRINT " DISKETTE":
     GOSUB 4:
     ON BC% GOTO 282

312  PRINT D$"CATALOG"DR$(DR):
     & N(PF$,1,P,N%,SF%):
     PRINT "[M]"SF%" SECTORS FREE[M][M]
        "N% - 1" PICTURE(S)[M][M]":
     PF$(0) = "0.  NONE":
     ON N% = 1 GOTO 314:
     FOR I = 1 TO N% - 1:
        PRINT I"." TAB( 5)PF$(I):
     NEXT I

314  PRINT "[M]"PF$(0)"[M]":
     & I(SR$,0;A,"38",0,1,A > = 0 AND A <N%):
     ON BC% GOTO 282,282,308

316  EF = 0:
     ON A = 0 GOTO 282:
     PRINT D$"BLOAD"PF$(A);DR$(DR):
     ON EF GOTO 316:
     HOME :
     VTAB 21:
     INVERSE :
     PRINT PF$(A);:
     NORMAL :
     PRINT " RETRIEVED[M]CTRL-T FOR TEXT (MENU);[M]
        CTRL-V FOR VIEWING"

318  & D(G):
     VTAB 3:
     PRINT "[M]DISPOSITION OPTIONS:[M][M]
        1. PRINT, IF EPSON WITH GRAPPLER[M]
        2. RETRIEVE ANOTHER PICTURE[M][M]
        0. NONE OF THESE[M][M]":
     & I(SR$,0;A,"39",BC%,1,A > = 0 AND A <3):
     ON BC% GOTO 308,282,308:
     ON A = 0 GOTO 282:
     ON A = 2 GOTO 312

320  PR# 1:
     PRINT  CHR$ (9);"GE":
     PR# 0:
     & D(T):
     GOTO 312
```

(322-324) Sort to identify elements with centroids inside the plot boundaries.

```
322  PRINT "[M]SORTING ELEMENTS...":
     & W((EC,1 TO NE,1),SL,SR,N%,BR%,ER%):
     & W((EC,BR% TO ER%,2),ST,SB,N%,BR%,ER%):
     NO = N%:
     IF N% >0 THEN
```

```
          & O(EC,BR% TO ER%,0,A):
         FOR I = 1 TO N%:
              N = I +  BR% - 1:
              XL(I) = EC(N,1):
              YL(I) = EC(N,2):
              I%(I) = EC(N,0):
    NEXT I

324  & O(EC,1 TO NE,0,A):
     HOME :
     RETURN
```

Sort to identify nodes inside plot boundaries

```
326  PRINT "[M]SORTING NODES...":
     & W((NC,1 TO NN,1),SL,SR,N%,BR%,ER%):
     & W((NC,BR% TO ER%,2),ST,SB,N%,BR%,ER%):
     & O(NC,BR% TO ER%,0,A):
     FOR I = 1 TO N%:
         A = BR% +  I - 1:
         XL(I) = NC(A,1):
         YL(I) = NC(A,2):
         I%(I) = NC(A,0):
     NEXT I:
     NO = N%:
     HOME :
     RETURN
```

(328-330) Calculate screen coordinates

```
328  HGR :
     & D(G):
     VTAB 21:
     POKE 34,20:
     HOME :
     PRINT "CALCULATING COORDINATES...":
     XA = SL - SC * UL:
     YA = SB +  SC * VL:
     & M(NC = X * (SC)):
     FOR I = 1 TO NN:
         NC(I,1) = XA +  NC(I,1):
         NC(I,2) = YA - NC(I,2):
     NEXT I

330  & M(EC = CC * (SC)):
     FOR I = 1 TO NE:
         EC(I,1) = XA +  EC(I,1):
         EC(I,2) = YA - EC(I,2):
     NEXT I:
     HCOLOR= 3:
     SCALE= 1:
     ROT= 0:
     HOME :
     & D(V):
     RETURN

332  PRINT D$"BLOAD SHAPES,A$193E,D1":
     PRINT D$"BLOAD HI-RES DUMP,A$1D00":
     RETURN
65535 REM [M][M]1FEB85[M]JRC/DCD

END-OF-LISTING
```

DISKETTE PREPARATION.HEAT

Cold start initialization.

```
1  TEXT :
   & D(T):
   WR$ = "...WHEN READY, PRESS <RETURN>":
   D$ =  CHR$ (4):
   FI$ = "FILEINFO.TXT":
   POKE 2164,7:
   REM  DISK PREP(H)
```

Title and abstract.

```
2  HOME :
   GOSUB 38:
   HTAB 12:
   FLASH :
   PRINT "HEAT CONDUCTION":
   NORMAL :
   HTAB 4:
   PRINT "DATA DISKETTE PREPARATION (7.0)":
   GOSUB 38:
   PRINT "[M][M]ABSTRACT: THE PROBLEM DESCRIPTION
      AND   DEFAULT DATA FILE NAMES WILL BE STORED  IN
      FILE '"FI$"'."
```

Request data diskette.

```
4  DR =  PEEK (2048):
   DR$(1) = ",D1":
   DR$(2) = ",D2":
   PRINT "[M]INSERT ";:
   FLASH :
   PRINT "DATA";:
   NORMAL :
   PRINT " DISKETTE TO BE PREPARED INTODRIVE "DR:
   & I("[M]<RETURN>";"";A$,"1"):
   ONERR  GOTO 40
```

Warn of diskette erasure.

```
6  & I("IF REFORMATTING, CATALOG? (Y/N)";"N";
      A$,"2",0,1,(A$ = "Y" OR A$ = "N")):
   IF A$ = "Y" THEN
      PRINT D$"CATALOG";DR$(DR):
      & I("CONTINUE? (Y/N)";"Y";A$,"3",0,1,
         (A$ = "Y" OR A$ = "N")):
      ON A$ = "N" GOTO 32
```

Prepare a data diskette having no DOS or greeting program. Get keyword and problem description.

```
8  PRINT D$"PREP"DR$(DR):
   POKE  - 16368,0:
   HOME :
   & C(NA$,DE$,D1$,D2$):
   DIM  NA$(22),DE$(22),D1$(22),D2$(22):
   PRINT "[M]SPECIFY:":
   & I(" KEYWORD:";"PROB 1";KW$,"4",BC%,8):
   ON BC% < >0 GOTO 2:
   HTAB 3:
   & I("[M] DESCRIPTION:","",DS$,"5",BC%,80):
   ON BC% < >0 GOTO 10
```

(10-24) Define default file information.

```
10  DATA "COORDINATES","COORDINATES OF INPUT DATA POINTS",
      "POINTS","DIRECTIONS","REGIONS",
      "NODE AND CONNECTIVITY DATA","REGIONS","NUMBERS EACH",
      "ELEMENT NODES","ELEMENT NODE NUMBERS","ELEMENTS",
      "NODES EACH"

12  DATA "NODE COORDS","COORDINATES OF NODES","NODES",
      "DIRECTIONS","RENUM EL ND",
      "RENUMBERED ELEMENT NODE NUMBERS","ELEMENTS",
      "NODES EACH","RENUM ND COORDS",
      "COORDINATES OF RENUMBERED NODES","NODES",
      "DIRECTIONS"

14  DATA "RENUM LINES","RENUMBERED NODES FOR UNIQUE LINES",
      "LINES","NODES/EL"

16  DATA "RENUM BOUNDS","RENUMBERED BOUNDARY NODES & ELEMENTS",
      "NODES","NODES/EL","PROPERTIES",
      "MATERIAL THERMAL PROPERTIES","ELEMENTS","PROPERTIES",
      "INPUT BC","INPUT BOUNDARY CONDITION SPECIFICATIONS"

18  DATA "INPUTS","PARAMETERS","BOUND COND",
      "NODAL BOUNDARY CONDITIONS","NODES","CODE/VALUE",
      "INIT SOURCES","INITIAL HEAT SOURCE VECTOR","NODES",
      "VALUE","INIT STIFF","INITIAL GLOBAL STIFFNESS MATRIX"

20  DATA "NODES","BANDWIDTH","CONV STIFFNESS",
      "STIFFNESS INCLUDING CONVECTION B.C.","NODES","VALUE",
      "MOD SOURCES","SOURCES MODIFIED BY TEMPERATURE B.C.",
      "NODES","VALUE","MOD STIFF",
      "STIFFNESS MODIFIED BY TEMPERATURE B.C."

22  DATA "NODES","BANDWIDTH","NODE TEMPS","NODAL TEMPERATURES",
      "NODES","VALUE","ELEMENT TEMPS",
      "AVERAGE ELEMENT TEMPERATURES","ELEMENTS","VALUES",
      "EL TEMP GRADS","TEMPERATURE GRADIENTS AT ELEMENTS"

24  DATA "ELEMENTS","COMPONENTS","ND TEMP GRADS",
      "TEMPERATURE GRADIENTS AT NODES","NODES","COMPONENTS",
      "ND HT SOURCES","NODAL EQUIVALENT HEAT SOURCES",
      "NODES","VALUES","BOUNDARY FLUXES",
      "HEAT FLUXES NORMAL TO BOUNDARIES","BOUNDS",
      "NODES/VALUE"
```


```
24  DATA "ELEMENTS","COMPONENTS","ND TEMP GRADS",
      "TEMPERATURE GRADIENTS AT NODES","NODES","COMPONENTS",
      "ND HT SOURCES","NODAL EQUIVALENT HEAT SOURCES",
      "NODES","VALUES","BOUNDARY FLUXES",
      "HEAT FLUXES NORMAL TO BOUNDARIES","BOUNDS",
      "NODES/VALUE"
```


(26-28) Define files to be created and prepare the 'fileinfo.txt' random access file.

```
26  PRINT "[M][M]CREATING "FI$:
    RESTORE :
    FOR I = 1 TO 22:
        READ NA$(I),DE$(I),D1$(I),D2$(I):
    NEXT I:
    N = 1:
    S = 0:
    PRINT D$;"OPEN"FI$",L100";DR$(DR):
    PRINT D$;"DELETE"FI$:
    PRINT D$;"OPEN"FI$",L100":
    PRINT D$;"WRITE"FI$",R0":
```

```
    PRINT N:
    PRINT KW$:
    PRINT DS$

28  FOR RE=1 to 22:
    PRINT D$"WRITE"FI$",R";RE:
    PRINT S:
    PRINT KW$ +  "/" +  NA$(RE):
    PRINT DE$(RE):
    PRINT N:
    PRINT D1$(RE):
        PRINT N:
        PRINT D2$(RE):
    NEXT RE:
    PRINT D$"CLOSE"FI$:
    PRINT "COMPLETED."
```

Review 'fileinfo.text' contents.

```
30  & I("REVIEW CONTENTS? (Y/N)";"N";A$,"6",0,1,(A$ = "Y" OR A$ = "N")):
    IF A$ = "Y" THEN
         PRINT "[M]USE CTRL S TO STOP/START LISTING":
        & B(10,10):
        & B(0,0,10):
        PRINT D$"TDUMP"FI$;DR$(DR):
        & I(WR$;"";A$,"7")
```

(32-36) Exit menu.

```
32  HOME :
    PRINT "DISK PREP: EXIT (7.10)[M][M]
       1. RETURN TO MAIN MENU[M]
       2. PREPARE ANOTHER DISKETTE[M][M]
       0. STOP[M][M]":
    & I("SELECT 0-2 ";1;A,"8",0,1,A > = 0 AND A < = 2):
    ON A GOTO 34,2:
    END

34  POKE 2165,1:
    ON DR = 2 GOTO 36:
    PRINT "[M]INSERT ";:
    FLASH :
    PRINT "PROGRAM";:
    NORMAL :
    PRINT " DISKETTE INTO DRIVE 1":
    & I(WR$;"";A$,"9")

36  PRINT D$"RUN HELLO,D1"

38  FOR I = 1 TO 40:
        PRINT "*";:
    NEXT I:
    RETURN
```

(40-44) Error traps.

```
40  POKE 216,0:
    ON  PEEK (222) = 255 GOTO 48:
    ON  PEEK (222) = 4 GOTO 44:
    ON  PEEK (222) = 8 GOTO 42:
    PRINT "DOS ERROR":
    END

42  PRINT "DOOR OPEN, I/O ERROR OR[M]NOT YET INITIALIZED":
    GOSUB 46:
    RESUME
```

```
44  PRINT "[G]DISK IS WRITE PROTECTED":
    GOSUB 46:
    RESUME
```

Wait for user response subroutine.

```
46  & B(20,20):
    & I("WHEN READY, PRESS RETURN <RETURN>";"";A$,"10"):
    RETURN

48  END

65535 REM [M][M]1FEB85[M]JRC/DCD

END-OF-LISTING
```

TEXT.HEAT

```
2  TEXT :
   & D(T):
   HOME :
   INVERSE :
   FOR I = 1 TO 4:
      PRINT  TAB( 2)" " TAB( 38)" ":
   NEXT I:
   NORMAL :
   VTAB 2:
   HTAB 3:
   PRINT " APPLIED FINITE ELEMENT ANALYSIS: ":
   HTAB 3:
   PRINT "     AN APPLE " CHR$ (221) CHR$ (219)
     " IMPLEMENTATION      ":
   VTAB 6:
   HTAB 12:
   PRINT "COPYRIGHT, 1985":
   PRINT

4  HTAB 12:
   INVERSE :
   PRINT "HEAT TEXT";:
   NORMAL :
   PRINT " (8.00)":
   PRINT "ABSTRACT:" PRINT "GRID, PROPERTY,
     AND B.C. DATA ARE READ FROM BINARY DATA FILES
     AND WRITTEN TO    TEXT FILES."

6  POKE 2166,8:
   FI$ = "FILEINFO.TXT":
   GOSUB 508:
   & C(DR$):
   DIM DR$(2):
   DR$(1) = ",D1":
   DR$(2) = ",D2":
   & B(200,20):
   PRINT "INSERT ";:
   FLASH :
   PRINT "DATA";:
   NORMAL :
   PRINT " DISKETTE INTO DRIVE "DR:
   GOSUB 508:
   ONERR  GOTO 510

8  PRINT D$"OPEN"FI$",L100"DR$(DR):
   PRINT D$"READ"FI$",R0":
   INPUT TE,KW$,PD$:
   PRINT D$"CLOSE"FI$:
   PRINT "PROBLEM KEYWORD: "KW$:
   PRINT :
   PRINT "PROBLEM DESCRIPTION :":
   PRINT PD$:
   GOSUB 508:
   & C(SI%):
   DIM SI%(22)

10 RE = 5:
   GOSUB 500:
```

```
    NE = N1:
    DE = N2:
    & C(ND%):
    DIM ND%(NE,DE):
    & R(ND%,N$):
    GOSUB 506:
    RE = 6:
    GOSUB 500:
    NN = N1:
    & C(XY):
    DIM XY(NN,2):
    & R(XY,N$):
    GOSUB 506:
    REM   " RETRIEVE NODES OF ELEMENTS (ND%)
      AND COORDINATES OF NODES (XY)

12  RE = 7:
    GOSUB 500:
    NL = N1:
    & C(LN%):
    DIM LN%(NL,3):
    & R(LN%,N$):
    GOSUB 506:
    RE = 8:
    GOSUB 500:
    NB = N1:
    & C(BN%):
    DIM BN%(NB,2):
    & R(BN%,N$):
    GOSUB 506:
    REM   " RETRIEVE NODES OF LINES (LN%)
      AND BOUNDARY NODES (BN%)

14  RE = 9:
    GOSUB 500:
    & C(MP):
    DIM MP(NE,N2):
    & R(MP,N$):
    GOSUB 506:
    REM  " RETRIEVE MATERIAL PROPERTIES OF ELEMENTS (MP)

16  REM " COL. 1 IS X-DIRECTION THERMAL CONDUCTIVITY;
      COL. 2 IS Y-DIRECTION THERMAL CONDUCTIVITY;
      COL. 3 IS ELEMENT THICKNESS.

18  RE = 11:
    GOSUB 500:
    NP = N1:
    & C(BC):
    DIM BC(NP,2):
    & R(BC,N$):
    H = BC(0,0):
    F = BC(0,1):
    GOSUB 506:
    REM  " RETRIEVE BOUNDARY CONDITIONS AT NODES
      (ROWS 1 TO NN OF MATRIX  BC).
      COL. 1 IS B.C. TYPE (1=SOURCE, 2=TEMP)

20  REM    " SURFACE CONVECTION B.C. ARE IN ROWS NN+ 1 TO NN+ H.
      COL. 0 IS SIDE NO.;
      COL. 1 IS ELEMENT NO.;
      COL. 2 IS PRODUCT OF CONVECTION COEFF.
      AND FLUID TEMPERATURE.
```

```
22  REM " FACE CONVECTION B.C. ARE IN ROWS NN+ H+ 1 TO NN+ H+ F.
       COL. 1 IS  ELEMENT NO.;
       COL. 2 IS PRODUCT OF CONVECTION COEFF.
       AND FLUID TEMPERATURE.

24  PRINT :
    PRINT "------ DATA FILES HAVE BEEN READ ----- ":
    PRINT :
    PRINT "  PREPARE TO TRANSFER":
    END :
    REM  "[M][M]ADD OUTPUT CODE TO TEXT FILE HERE[M][M]

500 PRINT D$"OPEN"FI$",L100"DR$(DR):
    PRINT D$"READ"FI$",R"RE:
    INPUT SI%(RE),NA$,DE$,N1,D1$,N2,D2$:
    PRINT D$"CLOSE"FI$:
    N$ = NA$ +  DR$(DR):
502 PRINT "*** ";
    INVERSE :
    PRINT "LOADING";:
    NORMAL :
    PRINT " DATA FROM DISKETTE ***":
    PRINT :
    PRINT "FILE NAME FOR": `
    & I(DE$,NA$,NA$,"1",BC%,30):
    ON BC% >0 GOTO 502:
    ON  LEFT$ (NA$, LEN (KW$)) = KW$ GOTO 504:
    NA$ = KW$ +  "/" +  NA$

504 N$ = NA$ +  DR$(DR):
    RETURN

506 PRINT :
    PRINT DE$:
    PRINT "FOR "N1" "D1$" AND "N2" "D2$:
    PRINT "WERE LOADED":
    RETURN

508 PRINT :
    & B(20,20):
    VTAB 24:
    & I("... WHEN READY, PRESS <RETURN>";"2";A$):
    HOME :
    RETURN

510 PRINT :
    FLASH :
    PRINT  CHR$ (7) CHR$ (7)"DATA NOT DEFINED":
    NORMAL :
    PRINT "EITHER FILE":
    INVERSE :
    PRINT NA$:
    NORMAL :
    PRINT "DOESN'T EXIST OR THE DATA IS NOT VALID":
    END

65535 REM [M][M]1FEB85[M]JRC/DCD

END-OF-LISTING
```

2.4.1 Variable Definitions and Cross References

```
*****************************
*                           *
*        HELLO.HEAT         *
*                           *
*    TABLE OF VARIABLES     *
*                           *
*****************************
```

A - GENERAL VARIABLE; FREQUENTLY ASCII VALUE OF INPUT CHARACTER

12 12 12 12 12 12 12 12 14
14 14 14 16

A$ - GENERAL STRING VARIABLE; FREQUENTLY INPUT STRING
10 12 16 16 20 38

B(*,*) - & BEEP COMMAND
12 12 30

B(*,*,*) - &BEEP (WITH REPETITION)
30

BC% - BRANCH CODE (1 ='<' BACKWARD; 2 ='>' FORWARD BRANCH; 3 =':' A PREVIOUS VALUE
14 14 20 20

D$ - CHR$(4) CONTROL D
4 6 8 8 36 36

D(*) - & DISPLAY (&D(T)=TEXT,&D(G)=MIXED,&D(V)=FULL GRAPHICS)
6

DR - DISKETTE DRIVE NUMBER
36

DR$(*) - DISKETTE DRIVE STRING
36

I - GENERAL INDEX
18 18 26 26 26 26 26 26 26
28

I(*,*) - & INPUT CMD
10 12 38

I(*,*,*) - & INPUT CMD
20

I(*,*,*,*) - & INPUT CMD
16

I(*,*,*,*,*) - & INPUT CMD
12 28

I(*,*,*,*,*,*) - & INPUT CMD
14

LI(*) - LINE LOCATION IN HELLO MENU
32

N - MENU OPTION
26 28 28 28 28 28 32 32 32
32 32 32 36

PD$(*) - PROBLEM DESCRIPTOR
26 26 26 26

PN$(*) - PROGRAM NAME OPTIONS
26 32 36

SR$ - 'SELECT A NUMBER' PROMPT
4 14 28

T - &D(T) SETS TEXT DISPLAY
6

WR$ - 'WHEN READY' PROMPT
4 10 12 20

1 FEB 85 Version JRC/DCD

END OF VAR. LIST

```
*****************************
*                           *
* GEOMETRY.ELASTICITY/HEAT  *
*                           *
*    TABLE OF VARIABLES     *
*                           *
*****************************
```

*A(*) - ANGLE (RADIANS, CCW) TO 1ST POINT
92 292

*EQ(*) - TESTS EQUIVALENCE OF COORDINATES OF 2 POINTS
12 94 94 102 292

*PX(*) - CONVERTS X ABSOLUTE COORDINATE TO SCREEN COORDINATE
10 10 14 18 26 28 28 30 30
32 34 50 66 120 120 122 142
142 202 206 206 212 218 276

*PY(*) - CONVERTS Y ABSOLUTE COORDINATE TO SCREEN COORDINATE
10 10 14 18 26 28 28 30 30
32 34 50 66 120 120 122 142
142 202 204 206 212 212 218
276

*RS(*) - ROUNDING FUNCTION FOR PRINTING REAL NUMBERS
208 210 210 214 214 214 278

*S(*) - STEPPING FUNCTION AROUND A

MATRIX OF NP VALUES (UP IF A=21; DOWN IF A=8)
10 66 122 122 144 292

*UX(*) - CONVERTS X SCREEN COORDINATE TO ABSOLUTE COORDINATE
210 276

*VY(*) - CONVERTS Y SCREEN COORDINATE TO ABSOLUTE COORDINATE
204 214 276

*WN(*) - STEPPING FUNCTION FOR INDEX OF REGION CONNECTIVITIES
42 48 140

A - GENERAL VARIABLE; FREQUENTLY ASCII VALUE OF INPUT CHARACTER
10 44 44 52 52 52 52 52 54
54 54 54 56 56 66 122 122
142 144 144 152 152 152 152
154 154 154 158 158 158 158
158 158 160 160 160 160 160
194 194 292 292 310 312 312

A$ - GENERAL STRING VARIABLE; FREQUENTLY INPUT STRING
4 10 10 10 10 66 66 66 96
96 108 108 108 108 108 108
108 110 110 110 110 110 110
110 110 120 122 122 122 122
124 128 128 128 128 130 130
130 130 130 130 130 142 142
144 144 154 154 162 162 168
168 178 178 258 258 260 260
294 294 296 296 312 312

A0 - GENERAL VARIABLE
70 70 70 76 76 76 76 94 94
94 94

A1 - GENERAL VARIABLE
24 70 92 92 120 120 120 120
122 122 132 132 132 132 148
148 148 174 176 176 176 176
176 176 176 178 178 178 178

AC - ACTION OPTIONS(1=ENTER;2=DELETE;3=QUIT)
108 108

AC$ - ACTION OPTION DESCRIPTORS
278

AC$(*) - STRING FOR ACTION OPTION (E=ENTER;D=DELETE;Q=QUIT)
278 280 280

AE - AN IN RADIANS; ANGLE FROM HORIZONTAL
222 292

AN - INPUT ANGLE (DEGREES); INCLUDED ANGLE
70 92 92 92 92

AR - 'AN' IN DEGREES; ANGLE FROM HORIZONTAL
92 92 92

B$ - MESSAGE ARRAY
126 312

B$(*) - MESSAGE ARRAY , E.G. "MARKING POINT"
118 118 118 120 310 310 310
312

B(*,*) - & BEEP COMMAND
4 6 76 94 120 122 272 288

B(*,*,*) - &BEEP (WITH REPETITION)
6 28 28 174

BC% - BRANCH CODE (1 ='<' BACKWARD; 2 ='>' FORWARD BRANCH; 3 =':' A PREVIOUS VALUE
4 40 40 54 54 60 60 62 62
72 72 72 74 74 76 76 76 80
80 80 82 82 84 84 84 84 84
86 86 86 86 86 88 88 88 88
88 90 90 90 92 92 96 96 96
96 100 100 100 100 106 106
108 108 110 110 116 118 118
130 130 138 138 140 148 148
152 152 152 152 152 152 152
154 154 158 158 158 158 160
162 162 166 166 166 166 166
168 168 172 172 178 178 178
226 226 234 234 274 274 274
284 284 286 286 288 288 294
294 296 296 296 298 298 298
298 300 300 300 300 302 302
304 304 304 316 318

C$ - GENERAL STRING VARIABLE
126

C$(*) - GENERAL STRING VARIABLE
118 118 120 120

C(*) - CLEAR ARRAY (&C(*))
180 186 194 194 196 266 312

C(*,*) - & CLEAR ARRAY CMD
42 126 294

C(*,*,*) - & CLEAR ARRAY (3 PARM)
140 270

C(*,*,*,*,*) - & CLEAR ARRAY (5 PARM)
278

C(*,*,*,*,*,*,*) - & CLEAR ARRAY (7 PARM)
276

CN% - CONNECTED NODES
42

GN%(*,*) - GLOBAL NODE NUMBERS (REGION, LOCAL NODE # 1-8)
16 24 24 26 26 28 28 28 30
32 32 32 36 38 40 40 40 42
42 46 46 48 48 48 122 124
124 142 142 142 142 158 158
160 174 174 186 198 198 278
308

GS% - GLOBAL NODE NUMBERS FOR CENTERS OF INPUT REGION SIDES
42

GS%(*) - GLOBAL NODE NUMBERS FOR CENTERS OF INPUT REGION SIDES
14 14 14 16 18 18 18 42 42
42 48 52 52

I - GENERAL INDEX
12 12 12 12 14 14 14 14 14
14 14 16 16 16 16 16 16 18
18 18 18 18 18 18 20 20 20
20 20 20 24 24 24 24 24 24
24 24 24 24 28 28 28 28 30
30 30 32 32 32 32 32 32 32
32 38 38 38 46 46 46 46 48
56 56 56 56 56 56 56 56 56
56 56 56 56 56 56 56 94 98
98 98 100 100 100 100 100 100
102 146 146 146 146 174 174
174 174 174 176 176 176 176
176 176 180 180 180 180 180
180 180 180 186 186 186 186
186 186 186 186 186 186 186
186 194 194 194 194 194 194
198 198 198 198 198 198 198
198 198 198 198 198 206 206
212 212 218 218 218 218 264
264 292 292 306 306 306 306
306 308 308 308 308 310 310
310 312 312 312 314 314 314
314 314

I$ - INPUT WITH/WITHOUT GRAPHICS STRING
270

I$(*) - INPUT WITH/WITHOUT GRAPHICS STRING
270 270 270 282 282 282 282
294

I(*,*) & INPUT CMD
124 160 258 260

I(*,*,*) - & INPUT CMD
4 80 82 152 152 296 298 300

I(*,*,*,*) - & INPUT CMD
96 162 294

I(*,*,*,*,*) - & INPUT CMD
40 54 60 62 72 74 86 88 92
96 100 108 110 118 128 130
154 154 158 162 168 168 178
226 298 300

I(*,*,*,*,*,*) - & INPUT CMD
166 166 172 234 274 284 286

I1 - GENERAL SUBSCRIPT
16 16 16 16 16 18 18 18 18
56 202 202 202 202

IG - INPUT WITH GRAPHICS (0,NO; 1, YES)
132 132 148 148 154 284 294

J - GENERAL SUBSCRIPT
24 24 24 24 24 36 36 36 40
40 40 40 124 124 124 158 158
158 158 158 158 158 158 174
174 174 174 174 174 176 176
176 176 186 186 186 186 186
186 186 186 198 198 198 198
198 198 198 198 308 308 308
310 310 312 312 312 312

JT% - CONNECTIVITY DATA (REGION, ADJACENT REGION BY SIDE)
276

JT%(*,*) - CONNECTIVITY DATA
44 44 46 48 174 174 174 176
186 198 278 310 312 312

K - GENERAL SUBSCRIPT
40 40 40 40 44 44 46 174 174
174 176

KW$ - KEYWORD
238 238 238 244 246 246 248

L$ - GENERAL LABEL STRING
32 32 32 32 202 202 202 202
202 204 204 204 204 204 208
210 210 210 210 214 214 214

L(*) - & LISTING CONTROL
1 4 6 282 306 308 310 314

L1$ - GENERAL STRING
204 204

LD - CODE FOR DEFINING REGIONS (1=POINT; 2=LINE; 3=ARC)
80 80 86 88 92 98 100 102
102 102 110 110

LD$ - DESCRIPTOR OF REGION INPUT CODES, LD
278

LD$(*) - REGION GENERATING OPTION STRING (POINT/LINE/ARC)
80 278 280 280 280

LG - NATURAL LOGARITHM OF X
278 278 278

LL - LINE LENGTH
204 204

LN - LOCAL NODE NUMBER
26 26 46 46 48 48

MO - MENU OPTIONS
112 116 284 284 284 284 284
284 284 294 294 294 296 296
296

N - SUBSCRIPT; COMMONLY IN STEPPING FUNCTION
8 8 8 8 10 10 10 10 10 10
20 20 22 22 22 22 24 24 26
26 26 34 34 36 36 38 38 38
38 42 42 42 42 42 42 42 44
44 44 44 44 46 48 48 48 48
48 48 48 48 48 48 48 56 64
66 66 66 66 66 66 66 66 66
66 68 68 70 70 72 72 72 74
74 74 76 76 76 76 78 84 84
86 86 86 86 86 88 88 88 90
90 90 90 92 92 92 92 94 94
94 94 94 98 98 100 100 102
102 102 102 120 120 120 120
122 122 122 122 122 122 122
124 124 124 124 124 124 140
140 140 140 142 142 142 142
144 144 144 144 292 292 292
292 292 292 292 292 292 292
292 292 292 292

N$ - GENERAL STRING
182 188 194 198 224 226 236

N0 - GENERAL INDEX
42 56 56

N1 - GENERAL INDEX; NUMBER OF ROWS IN SAVED ARRAY
28 28 28 28 28 28 30 30 30
30 30 42 42 42 42 44 44 44
44 44 48 48 48 48 48 68 68
68 68 72 74 76 76 78 78 78
78 86 90 90 90 92 92 92 92
94 102 102 102 180 186 194
198 224 224 228 228 242

N2 - GENERAL INDEX; NUMBER OF COLUMNS IN SAVED ARRAY
68 68 76 76 76 76 76 78 78
78 92 94 94 102 102 102 180
186 224 228 228 242

NA$ - FILENAME

200 224 224 226 226 226 228
234 234 236 238 238 238 240
258

NC% - NUMBER OF COLUMNS OF NODES (REGION)
140 276

NC%(*) - NUMBER OF COLUMNS OF NODES (REGION)
16 18 56 56 140 146 166 166
166 186 198 278 314

ND - NODE #
24 24 26 26 26 26 26 26 26
38 120 120 120 122 122 122
122

NF - NOT FULLY DEFINED FLAG
146 146 146 146

NI - NUMBER OF INTERMEDIATE POINTS ON AN ARC OR LINE; NUMBER OF INPUT VALUES
78 96 96 96 96 98 98 100 100
102

NM - NEXT MENU DEFAULT
282 284

NN - NUMBER OF NODES
140 144

NO - GENERAL SUBSCRIPT
12 64 76 90 94 150 152 152
152 152 152 152 152 152 152
152 152 152 154 154 154 156
158 158 158 160 160 160 160
160 162 162 162 162 164 166
166 166 166 166 166 166 166
166 166 166 166 166 168 168
168 168 172 172 172 172 172

NP - NUMBER OF DATA POINTS
10 10 12 12 12 20 20 20 20
22 22 22 22 22 60 60 60 60
62 62 62 62 64 64 64 64 66
66 66 66 72 72 80 80 86 86
86 86 88 102 102 122 122 152
152 152 154 158 180 180 180
194 194 194 194 194 194 194
218 218 292 292 292 292 292
294 306

NQ - NUMBER OF QUADRILATERAL REGIONS
40 40 40 40 40 56 118 118
126 126 126 126 144 144 144
146 160 160 160 160 160 162
166 168 174 174 174 176 176
176 186 186 186 198 198 198
220 220 294 308 310 314

NR% - NUMBER OF ROWS OF NODES (REGION)
140 276

NR%(*) - NUMBER OF ROWS OF NODES (REGIONS)
16 18 56 56 140 146 166 166
166 186 198 278 314

NS% - NUMBER OF NODES FOR SIDES GS%
140

NS%(*) - NUMBER OF NODES FOR SIDES GS%
42 42 46 48 48 52 52 140

O(*,*,*,*) - & ORDER CMD (SORT)
194 194

O1 - FLAG TO INDICATE DUPLICATE POINT (1=DUPLICATE; 2=NOT)
24 24 24 64 64 64 64 90 90
90 102

O2 - DUPLICATE POINT FLAG(1=DUPLICATE; 2=NOT)
24 24 24 24 76 76 76 94 94
94 94 96 102

OL - DUPLICATE POINT FLAG(1=DUPLICATE; 2=NOT)
12 12 12 12 64 76 90 94

P% - LISTING SPEED PARAMETER
1 306 308 310 314

PD$ - PROBLEM DESCRIPTOR
244 246 248

PN$ - PROGRAM NAME OPTIONS
276

PN$(*) - PROGRAM NAME OPTIONS
278 280 280 288 290

PO - PRINT OR PLOT OPTION
42 48 48 48 48

R$ - "RETAIN HOW MANY" STRING
1 160

R(*,*) - & RECALL ARRAY
194 198

R1 - REGION # OR RECORD NUMBER
184 190 250

R2 - REGION # OR RECORD #
184 190 250

RA - RADIUS OF ARC
70 70 92 92 92 94

RD - RADIANS/DEGREE CONVERSION
70 92 292 292 292

RD% - ARRAY CONTAINING REGION DEFINITION DATA
186 188 196 198

RD%(*,*) - ARRAY CONTAINING REGION DEFINITION DATA
52 52 186 186 186 186 186 186
198 198 198 198 198

RE - REGION # OR RECORD #
24 24 26 26 28 28 28 30 32
32 32 32 36 38 38 40 40 40
40 40 40 40 42 42 42 42 42
44 44 46 46 46 46 46 48 48
48 48 48 48 48 56 116 118
118 118 118 122 124 124 124
126 126 126 130 130 130 130
130 140 142 142 142 142 144
144 144 144 144 144 144 144
174 174 174 174 174 174 174
174 174 176 176 180 186 192
196 196 220 220 220 224 224
228 228 250 250 250 250

RS - REGION NUMBER
16 16 16 18 18 40 40 40 40
40 42 46 48 52 52 56

S - SCALE SIDE #
10 34 36 64 76 76 90 90 94
94 98 100 102

S(*,*) - & SAVE ARRAY CMD
182 188

S0 - SCALE FACTOR FOR FITTING WHOLE PLOT ON SCREEN
222 222 222 222 222

SB - SCREEN BOTTOM Y COORDINATE
208 212 214 216 222 276 276
276

SC - SCALE
222 222 222 222 222 222 276
276 276 276

SI% -STATUS INDICATOR ARRAY
278

SI%(*) - STATUS INDICATOR FOR RECORD(0 =UNDEFINED)
106 114 118 140 150 150 172
180 192 196 224 228 250 278
294 296 302 302 304 304 316
316

SL - SCREEN LEFT X COORDINATE
206 210 210 214 222 276 276
276

SR - SCREEN RIGHT X COORDINATE
208 210 210 210 222 276

SR$ - 'SELECT A NUMBER' PROMPT
1 172 274 284 286

ST - SCREEN TOP Y COORDINATE
214 214 214 216 222 276

SY - Y COMPONENT OF LINE SEGMENT
10 34 36 64 76 90 94 98 100
102

T - &D(T) SETS TEXT DISPLAY
1 104 114 132 134 148 156 164
170 180 188 282 318

TE - TYPE OF ELEMENT (1=TWO-DIMENSIONAL; 2=AXISYMMETRIC)
244 248 248 274 274 274 274
280 280 320 320

TN - TEMPORARY VALUE OF N
44 44 46 46 48 274 274 274
274 274 274 320

UE - ALLOWED ERROR IN COORDINATE VALUE
298 298 298

UH - HIGH VALUE OF X COORDINATE IN SPECIFIC PLOT
50 60 60 72 86 206 210 210
222 222 296 298 298 298 298
298 318

UL - LOW VALUE OF X COORDINATE IN SPECIFIC PLOT
50 60 60 72 86 206 208 208
212 214 222 222 276 276 296
298 298 298 298 298 318

UP$ - DESCRIBES INPUT OF POINT INSTRUCTIONS
270

UP$(*) - DESCRIBES INPUT OF POINT INSTRUCTIONS
72 80 86 270 270

VE - VOLUME OF ELEMENT
300 300 300

VH - HIGH VALUE OF Y COORDINATE IN SPECIFIC PLOT
50 62 62 74 88 212 214 214
222 222 296 300 300 300 300
300 318

VL - LOW VALUE OF Y COORDINATE IN SPECIFIC PLOT
50 62 62 74 88 206 208 210
212 214 214 222 222 276 276
296 300 300 300 300 300 318

WR$ - 'WHEN READY' PROMPT
1 4

X - FUNCTION DUMMY ARGUMENT
276 276 276 276 276 278 278
278 278 278 278 278 294

X(*) - WORKING ARRAY FOR X COORDINATES; GENERAL X COORDINATE
10 10 14 18 20 20 20 26 28
28 30 30 32 34 60 60 60 60
66 68 68 68 68 70 72 72 72
72 76 76 76 76 78 78 84 86
86 86 86 90 90 92 92 94 94
120 120 122 142 142 152 152
180 194 218 276 292 292 292
292 292 292 294 306

X1 - GENERAL X COORDINATE; START OF LINE
66 66 66 206 206 206 206 206
206 210 210 212 212 212 212
214

X2 - GENERAL X COORDINATE; END OF LINE
206 206 206 208 210 210 210
210 210 210

XA - X COORDINATE OFFSET FOR LABEL ON PLOTS
202 208 210 210 214

XC - GENERAL X COORDINATE; CENTROID COORDINATE
32 32 32 32 32 32 50 50 50
70 80 80 84 84 92 92 292 292
292 292

XH - MAX X COORDINATE
194 296 298 298 318

XL - MIN X COORDINATE

194 296 298 298 318

XP - GENERAL SCALED X COORDINATE
202 208 210 210 210 214

XY - COORDINATES OF POINTS
180 182 194 194 194 194 194

XY(*,*) - COORDINATES OF POINTS
180 180 180 180 194 194 194
194 194 194 194

Y - GENERAL Y COORDINATE
276 276 276 276 276 294

Y(*) - WORKING Y COORDINATE ARRAY; GENERAL Y COORDINATE
10 10 14 18 20 20 20 26 28
28 30 30 32 34 62 62 62 62
66 68 68 68 68 70 74 74 74
74 76 76 76 76 78 78 84 88
88 88 88 90 90 92 92 94 94
120 120 122 142 142 152 152
180 194 218 276 292 292 292
292 294 306

Y1 - GENERAL Y COORDINATE; START OF LINE
66 66 66 206 206 206 208 212
212 212 212 212 212 216 216
216

Y2 - GENERAL Y COORDINATE; END OF LINE
212 212 212 214 214 214 216
216 216 216

YA - Y COORDINATE OFFSET FOR LABEL ON PLOTS
202 208 214

YC - GENERAL Y COORDINATE; CENTROID COORDINATES
32 32 32 32 32 32 50 50 50
70 82 82 84 84 92 92 292 292

YH - MAX Y COORDINATE
194 296 300 300 318

YL - MINIMUM Y COORDINATE
194 296 300 300 318

YP - GENERAL SCALED Y COORDINATE
202 204 204 208 210 214 214
214 214

Z - USER INTERACTION INDICATOR (0, MAX; 1,INTERMEDIATE; 2,FREE RUN; 3,DEMO)
1 6 28 28 220 226 226 234
234 242 274 282

Z1 - LIMIT OF A VARIABLE
94 298

Z2 - LIMIT OF A VARIABLE
94 298

Z3 - LIMIT OF A VARIABLE
94 300

Z4 - LIMIT OF A VARIABLE
94 300

1 FEB 85 Version JRC/DCD

END OF VAR. LIST

```
*******************************
*                             *
*   GRID.ELASTICITY/HEAT      *
*                             *
*   TABLE OF VARIABLES        *
*                             *
*******************************
```

*P1(*) - STEPPING FUNCTION (CW) AROUND ELEMENT
24 254

*P2(*) - STEPPING FUNCTION (CCW) AROUND AN ELEMENT
24 40 40 254 324 324

*PX(*) - CONVERTS X ABSOLUTE COORDINATE TO SCREEN COORDINATE
6 6 6 6 8 12 16 18 56 68
68 68 68 100 246 248

*PY(*) - CONVERTS Y ABSOLUTE COORDINATE TO SCREEN COORDINATE
6 6 6 6 8 12 16 18 56 68
68 68 68 100 246 248

*S(*) - STEPPING FUNCTION AROUND A MATRIX OF NP VALUES (UP IF A=21; DOWN IF A=8)
8 56 58 62 62 254

*XC(*) - X COORDINATE OF ELEMENT CENTROID
58 60 60 114 246

*YC(*) - Y COORDINATE OF ELEMENT CENTROID
58 60 60 114 246

A - GENERAL VARIABLE; FREQUENTLY ASCII VALUE OF INPUT CHARACTER
8 8 10 14 16 24 24 24 26 36
36 36 42 42 44 44 44 48 56
58 62 64 64 64 64 70 72 110
120 120 120 126 126 126 134
134 144 144 180 198 198 254
254 284 286 286 286 322 322
322 322 322 322 324 324 324
324 326 332

A$ - GENERAL STRING VARIABLE; FREQUENTLY INPUT STRING
4 8 8 10 10 22 22 56 56 56
58 58 58 62 62 62 62 234 234
236 236 308 308 318 318 318
318 336 336 346 346

A1 - GENERAL VARIABLE
36 36 36

A1$ - GENERAL STRING
14 14 14 14 14 14 14 16

AN% -NODE NUMBERS FOR UNIQUE LINES
42 42 44 44 48 48 298 312
312

AN%(*,*) - ARRAY FOR NODE #S FOR UNIQUE LINES
40 40 40 40 42 44 44 44 54
54 54 54 164 164 164 298 298
312

B$ - MESSAGE ARRAY
104

B$(*) - MESSAGE ARRAY , E.G. "MARKING POINT"
104 104 106 110

B% - BANDWIDTH
36 36 36 36

B(*,*) - & BEEP COMMAND
4 36 120 126 168 248 264 268

B(*,*,*) - &BEEP (WITH REPETITION)
268 272

BC% - BRANCH CODE (1 ='<' BACKWARD; 2 ='>' FORWARD BRANCH; 3 =':' A PREVIOUS VALUE
4 18 18 18 18 38 98 138 140
150 156 190 190 192 194 194
196 196 204 204 214 214 258
258 262 262 264 264 270 270
276 276 278 278 278 280 280
282 282 288 290 290 294 296
296 308 312 314 318 320 320
320 320 322 322 322 322 322
322 332 334 336 336 336 338
338 338 338 342 342 346 346
346 346 346 346 348 348 350
352 352

BI - NODE # FOR NODE RENUMBERING
36 172 172 176 176 176

BL% - # BOUNDARY LINES
40 46 50 52

BL%(*,*) - BOUNDARY LINES ARRAY
44 44 44 46 46 50 50 50 50
52

BN% - BOUNDARY NODES
40 304 314 314

BN%(*,*) - BOUNDARY NODES ARRAY WITH NODES IN ORDER AROUND BOUNDARIES
44 46 46 46 46 46 50 50 50
50 166 166 314

BR% - BEGINNING ROW NUMBER IN &W()
24 24 24 26 26 28 28 42 42
42 42 44 44 44 44 44 46 46
50 50 50 50 50 50 52 52

BW% - BANDWIDTH
36 36 36 36 110 110 110 110
120 120 132 134 134 142 162
172 176 182 344 344 352 352

C(*) - CLEAR ARRAY (&C(*))
16 22 24 62 64 104 120 124 130
138 140 146 150 152 198 198
262 312 314 324 328

C(*,*) - & CLEAR ARRAY CMD
40 134 134 144 154 180 298

C(*,*,*) - & CLEAR ARRAY (3 PARM)
142 274 336

C(*,*,*,*) - & CLEAR CMD
168

C(*,*,*,*,*,*) - & CLEAR ARRAY CMD
152

C(*,*,*,*,*,*,*) - & CLEAR ARRAY (7 PARM)
186 286

C(*,*,*,*,*,*,*,*,*) - & CLEAR ARRAY CMD
286

C(*,*,*,*,*,*,*,*,*,*,*,*,*) - & CLEAR ARRAY CMD
118

CN% - RELATED NODES FOR GRID MODIFICATION
62 64 64

CN%(*,*) - TEMPORARY ARRAY OF RELATED NODES DURING GRID MODIFICATION
64 64 64 64 64 64 64 64 66
66 66 66 66 66 68 68 70 70
72 72 72 72 72 72

D - LOGICAL VARIABLE
48

D$ - CHR$(4) CONTROL D
1 202 202 202 206 206 206 212
212 224 224 224 234 236 250
250 250 266

D(*) - & DISPLAY (&D(T)=TEXT,&D(G)=MIXED,&D(V) =FULL GRAPHICS)
118 132 172 188 190 192 194
196 240 256 332 334 336 352
352

D1 - LENGTH OF DIAGONAL ACROSS QUADRILATERAL
104 106 336 336 340 340

D1$ - ROW DESCRIPTOR FOR SAVED DATA
202 206 208 222

D2 - LENGTH OF DIAGONAL ACROSS QUADRILATERAL
104 106

D2$ - COLUMN DESCRIPTOR FOR SAVED DATA
202 206 208 222

DA - INCREMENT IN ETA
76 78

DE - # NODES / ELEMENT
246 254 254 254 340

DE$ - DESCRIPTION OF DATA IN FNAMES
202 204 206 208 214 222

DF$(*) - COORDINATE (DIRECTION) DESCRIPTOR
8 8 14 14 18 18 112 112 160
160 192 192 254 254 254 320
320

DI$(*) - GENERAL COORDINATE (DIRECTION) DESCRIPTOR
254 254 254 254 254 254 254

DR - DISKETTE DRIVE NUMBER
202 206 212 218 224 234 236
248 248 250 264

DR$(*) - DISKETTE DRIVE STRING
202 206 212 218 224 234 236
244 248 248 250

DS - INCREMENT IN PSI
76 78

E - EXPONENTIAL (NOT A VARIABLE)
324

E0 -ERROR FLAG
212 212 238

E1 - ELEMENT NUMBER
60 60 60 60 62 64 66 66 72

E2 - ELEMENT NUMBER
60 60 60 60 64 66 66

EF - ERROR FLAG (1=ERROR OCCURRED; 0=NONE)
122 124 128 130 138 138 140
140 150 150 152 152 226 300
302 304 306 312 312 314 314

EL - ELEMENT # (GENERAL)
40 40 40 44 54 54 180 180
180 180 180 180

EN% - NODE # OF ELEMENT
120 122 124 138 138 144 180

EN%(*,*) - ELEMENT NODE #S (ELEMENT#, LOCAL NODE #)
120 120 120 138 142 142 180
180 180 182

EO - EXIT OPTION (1=PROCEED TO NEXT PROGRAM; 2=TO MAIN MENU; 3=REMAIN IN SAME; 0=STOP)
256 262 262 262 262 262 264 266

ER - ERROR CODE
226 226 226 226 226 226 226
226 226 226 226 226 226 226
234 234 234 234 234 234 234

ER$(*) - ERROR CODE STRING DESCRIPTORS
234 244 244 244 244 244 244
244

ER% - ENDING ROW NUMBER RETURNED FROM A VARIABLE SEARCH
24 24 24 24 26 42 42 42 44
44 46 50 50 52

ET - NORMALIZED COORDINATE FOR REGION TRANSFORMATIONS
78 78 78 78 78 78 78 78 78
78 80 80 80

F% - NUMBER OF VARIABLES FOUND IN A SEARCH
24 24 24 26 26 28 42 42 42
42 42 42 44 44 46 46 50 50 52

FI$ - 'FILEINFO.TXT'
202 202 202 206 206 206 212
224 224 224 246 250 250 250

FR - FREE RUN FLAG (1=FREE RUN; 0=USER CONTROL)
4 4 4 4 118 132 132 160 164
204 208 214 222 246 250 264
270 270 270 270 270 272 272
278 280 280 282 284 284 284
288 288 290 294 294 296 308
310 318 326 326 326

G - PARAMETER IN &D(G) SETS MIXED TEXT AND GRAPHICS
188 192 196 332 334

GM - FLAG FOR MODIFYING AN EXISTING GRID (1=YES; 0=NO)
276 276 276 276 276 276 276
278

GM$ - GENERATE MESH MESSAGES
274

GM$(*) - MESH GENERATION MESSAGES/ PROMPTS
274 274 274 274 276 276 276
276

GN% - GLOBAL NODE NUMBERS
152

GN%(*,*) - GLOBAL NODE NUMBERS (REGION, LOCAL NODE # 1-8)
76 82 82 154 154 198

HE$ - GENERAL HEADING
48 160 160 162 162 164 164
308 316

I - GENERAL INDEX
12 12 12 12 16 16 16 16 22
22 22 22 24 24 24 24 24 26
26 26 26 30 30 30 30 30 32
32 36 38 38 38 44 44 44 44
46 46 50 64 64 64 64 64 64
64 64 64 64 64 64 64 64 66
70 72 72 72 72 76 76 76 76
76 78 78 80 80 80 80 80 80
80 82 82 82 82 82 82 82 82
82 82 84 86 88 90 92 96 96
96 96 96 96 96 96 96 96 96
96 96 96 96 96 98 98 98 98
100 100 100 100 100 100 100
100 100 100 104 104 104 104
104 104 104 104 104 104 104
116 120 120 120 120 134 134
134 142 142 142 142 142 142
142 142 142 142 142 142 146
146 146 146 146 146 146 146
154 154 154 154 154 154 154
154 154 154 154 154 154 154
154 154 160 160 160 160 160
162 162 162 162 164 164 164
164 166 166 166 166 166 170
170 170 170 170 178 178 178
178 180 180 180 180 180 180
186 186 186 198 198 198 198
198 240 240 254 254 254 254
254 254 284 284 284 284 298
298 298 298 326 326 326 326
328 328 328 328 328 328 328
328

I(*,*,*) - & INPUT CMD
320 320 322 338 338

I(*,*,*,*) - & INPUT CMD
4 234 236 308 318 318 336 346

I(*,*,*,*,*) - & INPUT CMD
18 18 190 194 322 346 348

I(*,*,*,*,*,*) - & INPUT CMD
204 214 258 262 270 276 278
280 282 290 296 342 352

I1 - GENERAL SUBSCRIPT; STARTING SUBSCRIPT FOR A BOUNDARY
30 158 158 158 158 172 176

I2 - GENERAL SUBSCRIPT; ENDING SUBSCRIPT FOR A BOUNDARY
30 172 176

IC - CONNECTIVITY INDEX VALUES FOR GRID GENERATION
118 286

IC(*,*) - CONNECTIVITY INDEX VALUES FOR GRID GENERATION
82 286

II - SUBSCRIPT
76 76 76

IJ - SUBSCRIPT
106 106 106 108 108 116

IK - SUBSCRIPT
112 112 112 112 112 112 112
114

J - GENERAL SUBSCRIPT
24 24 24 24 24 24 24 24 24
24 24 24 24 34 34 34 36 36
36 36 40 40 40 40 40 40 40
40 40 44 44 44 44 64 64 64
64 78 78 80 80 80 80 80 80
80 82 82 82 82 84 86 88 90
92 94 96 96 96 98 98 98 100
100 100 100 100 100 100 100
100 100 104 104 104 104 104
104 104 104 104 104 104 116
120 120 120 120 134 134 134
134 142 142 142 142 154 154
154 154 154 154 154 154 162
162 162 162 170 170 170 170
180 180 180 180 198 198 198
298 298 298 298 322 322 322
322 322 322 322 322 322 322
324 324 324 324 324 324 324
324 324 324 324 324

J0 - INDEX FOR NODE WITHIN AN ELEMENT
112 112 112 112 112 112 112

J1 - GENERAL SUBSCRIPT
6 6 6 6 6 24 28 106 108 110
112

J2 - GENERAL SUBSCRIPT
6 6 6 24 28 106 108 110 112

J3 - GENERAL SUBSCRIPT
6 6 6 106 108 110 112

JK - GENERAL SUBSCRIPT
82 82 82 94

JL - GENERAL SUBSCRIPT
82 86 88 90 92 94 94

JT% - CONNECTIVITY DATA (REGION, ADJACENT REGION BY SIDE)
152

JT%(*,*) - CONNECTIVITY DATA
82 82 82 86 88 90 92 154 154

K - GENERAL SUBSCRIPT
28 28 28 28 30 30 30 30 34
34 36 36 36 64 72 72 72 72
72 72 80 80 80 80 80 80 82
82 84 100 100 100 100 100 100

K$ - STRING FOR COORDINATES OF NODES/NODES OF ELEMENTS STRING
274 342 344 346

O(*,*,*,*) - & ORDER CMD (SORT)
26 44 48 48 64 110 134 134
144 144 180 198 198

OL% - OLD NODE #S
168 186

OL%(*) - OLD NODE #S
30 30 34 34 34 34 168 180
180

P% - LISTING SPEED PARAMETER
32 98 104 160 164 170 178 240

P0 - PRINT OPTION
176 178

PD$ - PROBLEM DESCRIPTOR
250 250

PN$ - PROGRAM NAME OPTIONS
262

PN$(*) - PROGRAM NAME OPTIONS
264 264 264 264 266

PO - PRINT OR PLOT OPTION
22 30 32 32 34 36 38 48 48
76 98 100 102 104 110 110 110
112 114 114 116 118 170 176
176 178 178 280 280 280 282
282 282 284 284 288 288 290
290 290 294 294 296 296 296
326 326

Q$ - QUIT/LEFT ARROW/RIGHT ARROW PROMPT
1 56 58

R - ARRAY USED TO GENERATE ELEMENTS
118 286

R% - RELATED NODE NUMBERS FOR ONE NODE
16 22 24 24 26

R%(*) - RELATED NODE #
12 12 16 16 22 24 24 24 24
24 26 26 28 28

R(*) - USED TO GENERATE ELEMENTS
104 104 104 104 104 104 106
106 106 108 108 108 286

R(*,*) - & RECALL ARRAY
138 140 150 152 312 314

R1 - REGION # OR RECORD NUMBER
120 126 182 224 302

RB% - NODE NUMBERS USED IN GRID GENERATION
118 286

RB%(*,*,*) - NODE NUMBERS USED IN GRID GENERATION
86 88 90 92 96 96 96 96 286

RD - REDIMENSIONING FLAG
138 138 140 140 256 340 340
340

RD% - ARRAY CONTAINING REGION DEFINITION DATA
152 152 154

RD%(*,*) - ARRAY CONTAINING REGION DEFINITION DATA
152 154 154 154 154

RE - REGION # OR RECORD #
120 126 136 136 138 138 148
150 174 174 174 174 182 184
202 202 206 206 224 224 224
224 298 302 310 312 312

RN - RENUMBERED NODES FLAG (1=YES; 0=NO)
136 138 288 294 308 318 336
350

RN% - RELATED NODES
168 186

RN%(*,*) - RELATED NODES
22 34 168 170

S$ - 'STARTING'
1 8 62

S(*,*) - & SAVE ARRAY CMD
122 128 300 304

S0 - SCALE FACTOR FOR FITTING WHOLE PLOT ON SCREEN
200 200 200 200

S1 - VAR IN GRID GEN.
82 92 96 96

S2 - GRID GEN. VAR
82 88 96 96

SB - SCREEN BOTTOM Y COORDINATE
16 200 248 248

SC - SCALE
16 16 200 200 200 200 200 248
248

SH - SHAPE FUNCTIONS FOR REGION TRANSFORMATION
118 286

SH(*) - SHAPE FUNCTION
78 78 78 78 78 78 80 80 80
80 286

SI - PSI NORMALIZED FOR REGION TRANSFORMATION)
78 78 78 78 78 78 78 78 78
78 80 80 80

SI%(*) - STATUS INDICATOR FOR RECORD(0 =UNDEFINED)
136 138 148 150 186 202 206
224 244 278 278 294 310 310
312

SL - SCREEN LEFT X COORDINATE
16 200 248 248

SO - OUTPUT OPTION
278 278 278 280 280 280 284
326

SR - SCREEN RIGHT X COORDINATE
200 248

SR$ - 'SELECT A NUMBER' PROMPT
1 258 262 270 276 278 280 282
290 296 342 352

ST - SCREEN TOP Y COORDINATE
200 248

T - &D(T) SETS TEXT DISPLAY
118 132 172 190 194 240 256
336 352 352

TE - TYPE OF ELEMENT (1=TWO-DIMENSIONAL; 2=AXISYMMETRIC)
250 254 254

UH - HIGH VALUE OF X COORDINATE IN SPECIFIC PLOT
18 144 198 200

UL - LOW VALUE OF X COORDINATE IN SPECIFIC PLOT
16 18 144 198 200 248

VH - HIGH VALUE OF Y COORDINATE IN SPECIFIC PLOT
18 144 198 200

VL - LOW VALUE OF Y COORDINATE IN SPECIFIC PLOT
16 18 144 198 200 248

W% - BANDWIDTH
30 36 36 36 36 36 110 110
110 110

W(*,*,*,*,*,*) - & WHICH CMD (SELECTS ROWS)
24 24 26 42 42 44 46 50 52

X - FUNCTION DUMMY ARGUMENT
152 248 248

X(*) - WORKING ARRAY FOR X COORDINATES; GENERAL X COORDINATE
76 154 154 198

X1 - GENERAL X COORDINATE; START OF LINE
12 12 68 68 68 68 68 68

X2 - GENERAL X COORDINATE; END OF LINE
68 68 68 68

XA - X COORDINATE OFFSET FOR LABEL ON PLOTS
100 114 158

XC - GENERAL X COORDINATE; CENTROID COORDINATE
118 286

XC(*,*) - X COORDINATE OF NODE GENERATED FOR AN INPUT REGION
80 80 80 100 100 104 104 104
104 286

XE - X COORDINATE FOR ELEMENT
118 286

XE(*) - X COORDINATE FOR ELEMENT
100 112 112 286

XN - X COORDINATE OF NODE
142 186 286 336

XN(*) - X COORDINATE OF NODE
6 6 6 6 8 8 12 14 16 16 18
18 18 18 18 56 56 68 68 68
68 112 142 142 146 160 180
192 246 246 246 286 320 320
324 328 340

XP - GENERAL SCALED X COORDINATE
8 8 8 12 12 14 14 14 14 16
16 18 18 18 56 56 56 58 58
58 60 60 62 100 114 158

XR - X COORDINATES FOR LOCAL NODE #S IN A REGION
118 286

XR(*) - X COORDINATES FOR LOCAL NODE #S IN A REGION
76 76 76 80 286

XY - COORDINATES OF POINTS
150 150 154 198 198 198 198
324

XY(*,*) - COORDINATES OF POINTS
150 154 154 198 198 198 198
198 198 198 324 324 324 324
324 324 324

Y - GENERAL Y COORDINATE
152 248 248

Y(*) - WORKING Y COORDINATE ARRAY; GENERAL Y COORDINATE
76 154 154 198

Y1 - GENERAL Y COORDINATE; START OF LINE
12 12 68 68 68 68 68 68

Y2 - GENERAL Y COORDINATE; END OF LINE
68 68 68 68

YA - Y COORDINATE OFFSET FOR LABEL ON PLOTS
100 114 158

YC - GENERAL Y COORDINATE; CENTROID COORDINATES
118 286

YC(*,*) - GENERAL Y COORDINATE; CENTROID COORDINATES
80 80 80 100 100 104 104 104
104 286

YE - Y COORDINATE FOR ELEMENT
118 286

YE(*) - Y COORDINATE FOR ELEMENT
100 112 112 286

YN - Y COORDINATE OF NODE
142 186 286 336

YN(*) - Y COORDINATE OF NODE
6 6 6 6 8 8 12 14 16 16 18
18 18 18 18 56 56 68 68 68
68 112 142 142 146 160 180
192 246 246 246 286 320 320
324 328 340

YP - GENERAL SCALED Y COORDINATE
8 8 8 12 12 14 14 14 14 16
16 18 18 18 56 56 56 58 58
58 60 60 62 100 114 158

YR - Y COORDINATES FOR LOCAL NODE #S IN A REGION
118 286

YR(*) - Y COORDINATES FOR LOCAL NODE #S IN A REGION
76 76 76 80 286

Z - USER INTERACTION INDICATOR (0=MAX; 1=INTERMEDIATE; 2=FREE RUN; 3=DEMO)
1 36 132 214 222 256 258 258
270

1 FEB 85 Version JRC/DCD

END OF VAR. LIST

```
*******************************
*                             *
*      PREPROCESS.HEAT        *
*                             *
*    TABLE OF VARIABLES       *
*                             *
*******************************
```

*CX(*) - CALCULATES X SCREEN COORDINATE OF ELEMENT CENTROID
26 48 54 56 226 302 306 336

*CY(*) - CALCULATES Y SCREEN COORDINATE OF ELEMENT CENTROID
26 48 54 56 226 302 306 336

*E(*) - NODE NUMBER OR ELEMENT NUMBER
10 14 222

*L(*) - ELEMENT NUMBER OR INDEX IN ARRAY
10 14 222

*LE(*) - DEFINES LENGTH OF LINE
30 44 158 232

*M(*) - STEPPING FUNCTION AROUND BOUNDARIES
8 22 22 24 30 42 50 158 158
224

*O(*) - STEPPING FUNCTION AROUND ARRAY OF NODES
8 26 224

*P2(*) - STEPPING FUNCTION (CCW) AROUND AN ELEMENT
54 224 334

*PX(*) - CONVERTS X ABSOLUTE COORDINATE TO SCREEN COORDINATE
8 8 44 46 50 50 54 54 56
156 226 232 300 300

*PY(*) - CONVERTS Y ABSOLUTE COORDINATE TO SCREEN COORDINATE
8 8 44 46 50 50 54 54 56
156 226 232 300 300

*SI(*)
32 224

A - GENERAL VARIABLE; FREQUENTLY ASCII VALUE OF INPUT CHARACTER
8 22 24 26 30 42 50 82 82
82 82 82 82 82 82 84 84 84
84 84 84 84 86 86 86 90 90
90 90 90 90 90 90 92 92 102

102 102 102 102 102 102 102
104 104 104 104 104 104 104
106 106 106 106 152 152 152
152 158 158 162 166 166 224
224 224 224 300 334 334 334

A$ - GENERAL STRING VARIABLE; FREQUENTLY INPUT STRING
4 8 8 8 8 18 20 20 26 26
26 26 28 40 40 66 66 82 82
82 84 88 88 102 102 102 104
104 114 114 114 114 114 114
114 116 116 116 116 116 116
118 118 120 120 120 120 120
120 122 122 122 122 122 122
124 124 124 132 132 132 132
132 132 134 134 134 134 136
136 208 208 236 236 236 256
256 268 268 276 276 304 306

A0 - GENERAL VARIABLE
10 10 58 58 58 58 106 106
106 106 132 132

A1 - GENERAL VARIABLE
30 32 36 36 90 90

AC - INDEX FOR ACTION OPTIONS(1=ENTER; 2=DELETE; 3=QUIT)
18 20 22 22 24 28 114 116
118 122 124 126 128 132

AC$(*) - STRING FOR ACTION OPTION (E=ENTER; D=DELETE; Q=QUIT)
116 118 122 124 132 230 234
234

B - BOUNDARY NUMBER
8 22 22 24 30 42 50 50 50
50 50 50 50 112 112 112 134
134 134 134 134 158 158 158
158 158 158 224

B(*,*) - & BEEP COMMAND
4 6 82 102 220 252

B(*,*,*) - & BEEP (WITH REPETITION)
6

B0 - GENERAL VARIABLE
50 50

B1 - GENERAL VARIABLE
50 50 158 158 158

B2 - GENERAL VARIABLE
50 50 158 158 158

BC - EQUIVALENT BOUNDARY CONDITION MATRIX
144 148 150 326 328 332

BC% - BRANCH CODE (1 ='<' BACKWARD; 2 ='>' FORWARD BRANCH; 3 =':' A PREVIOUS VALUE
4 70 70 70 70 72 72 72 76
76 78 78 78 78 78 82 82 82
84 84 84 84 86 86 86 86 88
88 90 92 92 92 94 96 102 102
102 104 104 104 104 106 112
114 114 116 116 120 120 122
122 124 124 126 126 126 126
128 128 132 132 134 162 164
170 172 174 180 180 188 188
244 244 248 248 252 256 256
276 276 290 290 294 294 296
296

BC(*,*) -EQUIVALENT BOUNDARY CONDITIONS
32 32 32 32 32 32 32 32 32
34 34 34 36 36 36 38 38 38
144 146 316 316 318 318 318
326 328 328 330 330 334 334
336

BD - CODE FOR BOUNDARY CONDITION DIRECTION (1=X-DIRECTION; 2=Y-DIR; 3=NORMAL TO BOUNDARY (+OUT); 4=NORMAL TO FACE (SCREEN))
10 14 30 30 32 32 32 42 42
44 44 44 44 44 56 56 56 56
56 56 118 118 120 122 124 126
128 130 130 222 222 222 222
312 312 312 312 312 312 330
330

BD$(*) - STRING FOR BD
118 118 122 122 122 124 126
128 230 234 234 234 234 236
312

BI - NODE # FOR NODE RENUMBERING
138 140 142 278 280 280 312
322

BI% - MATRIX OF INPUT B.C. CODES
274 282 320 322 322

BI%(*,*) - MATRIX OF INPUT BOUNDARY CONDITION CODES (COL 1: INDEX IN BN%, COL 2: BC TYPE; COL 3 DIRECTION)
10 10 10 12 12 12 14 14 14
30 30 34 34 36 42 42 42 42
60 60 60 60 138 144 144 144
144 144 146 146 274 280 320
322 322

BI(*,*) - COMBINED BI% AND BV
138 138 138 138 278 280 280
280 312 312 312 312 322 322
322

BN% - BOUNDARY NODES
170 172

BN%(*,*) - BOUNDARY NODES ARRAY WITH NODES IN ORDER AROUND BOUNDARIES
8 30 30 30 42 42 50 50 50
50 154 156 158 158 170 222
312

BR% - BEGINNING ROW NUMBER IN &W()
302 302

BT - TYPE OF B.C. (0=INSULATED; 1=NODE HEAT SOURCE; 2=NODE TEMPERATURE; 3=SURFACE HEAT FLUX; 4=SURFACE CONVECTION)
10 14 14 20 20 22 30 30 32
34 34 34 38 42 42 42 42 42
42 42 44 46 46 46 48 48 48
118 118 118 118 118 118 118
122 124 124 126 126 128 128
130 144 144 146 146 146 146
146 222 222 312 312 312 312
312 330 330 330

BT$(*) - STRING FOR BT
116 116 116 116 118 122 124
126 128 128 230 236 236 236
236 236 312 316

BV - BOUNDARY VALUE
274 282 320 322 322

BV(*,*) - BOUNDARY VALUE
12 12 12 14 14 32 32 34 36
38 42 138 138 274 280 280 320
322 322 322

C(*) - CLEAR ARRAY (&C(*))
138 142 144 154 156 160 166
168 170 222 250 256 258 278
280 290 298 312 322 326 332

C(*,*) - & CLEAR ARRAY CMD
150 218 268 282 296 320 322
322

C(*,*,*) - & CLEAR ARRAY (3 PARM)
162 274

CO%(*) - CODES FOR COLOR OF FILL
54 228 302

CR - SCREEN COORDINATES OF INPUT REGION CENTROID
154

CR(*,*) - SCREEN COORDINATES OF REGION CENTROIDS
100 100 154 154 154 154 154
154 154 156 156 156 156 156
156

CV - CODED VALUE FOR MATERIAL PROPERTIES
228

CV(*,*) - CODED VALUES FOR MATERIAL PROPERTIES
78 78 84 98 104 228

D$ - CHR$(4) CONTROL D
1 40 178 178 178 182 182 182
186 186 198 198 198 200 200
200 208 254 266 266 266 266
266 274 274 274

D(*) - & DISPLAY (&D(T)=TEXT,&D(G)=MIXED,&D(V)=FULL GRAPHICS)
50 86 106 110 136 212 238 298

D1$ - ROW DESCRIPTOR FOR SAVED DATA
178 182 182 196

D2$ - COLUMN DESCRIPTOR FOR SAVED DATA
178 182 182 196

DB - DIMENSION FOR INPUT BOUNDARY CONDITIONS
274 274 320 320 322 322

DE - # NODES / ELEMENT
36 54 154 160 160 224 224

DE$ - DESCRIPTION OF DATA IN FILEINFO.TXT
178 180 182 182 188 196

DF$(*) - COORDINATE (DIRECTION) DESCRIPTOR
230 234 234 234 234 234 234

DI$(*) - GENERAL COORDINATE (DIRECTION) DESCRIPTOR
52 52 230 234 234 234 234 236
236

DR - DISKETTE DRIVE NUMBER
40 178 178 180 182 186 190
194 198 200 208 218 218 252
266 274

DR$ - DISKETTE DRIVE STRING
218

DR$(*) - DISKETTE DRIVE STRING
40 178 178 180 182 186 190
194 198 200 208 218 218 218
266 274

E$ - STRING 'ENDING'
1 20

E0 - ELEMENT NUMBER
74 90 90 90 252

E1 - ELEMENT NUMBER
68 80 90

E2 - ELEMENT NUMBER
68 80 86

E3 - ERROR FLAG FOR DISK I/O
186 186 210

ED - EDIT FLAG (1=EDIT MODE; 0=NOT)
94 108 112 136 136 256 256
262 276 276

EF - ERROR FLAG (1=ERROR OCCURRED; 0=NONE)
140 142 148 150 162 162 164
164 170 170 172 172 202 260
260 264 266 280 280 328 328

EL - ELEMENT # (GENERAL)
26 26 26 26 26 26 28 28 28
28 28 28 30 30 32 36 36 36
36 36 38 48 48 54 54 54 54
54 56 56 60 74 74 74 74 74
80 80 80 82 84 84 84 84 84
84 86 86 86 90 90 90 90 100
100 100 154 154 154 154 222
222 224 224 224 226 226 226
226 226 226 226 226 286 286
286 286 302 302 302 334 334
334 336 336 336

EO - EXIT OPTION (1=PROCEED TO NEXT PROGRAM; 2=TO MAIN MENU; 3=REMAIN IN SAME; 0=STOP)
80 238 248 248 248 248 248
252 254 254 254 254 254 254
254

ER - ERROR CODE
202 202 202 202 202 202 202
202 202 202 202 202 202 202
208 208 208 208 208 208 208

ER$ - ERROR CODE STRING DESCRIPTORS
218

ER$(*) - ERROR CODE STRING DESCRIPTORS
208 218 218 218 218 218 218
218

ER% - ENDING ROW NUMBER RETURNED FROM A VARIABLE SEARCH
302 302 302 302

F - &F FILL; COUNTER OF # BOUNDARY CONDITIONS ON THE FACE OF ELEMENTS (AS SEEN ON SCREEN)
38 38 38 38 144 144 144 144
146 316 328 330 330

F$ - STRING VARIABLE ="4."
122 236

F(*,*,*,*) - & FILL CMD
54 302 306

FA$ - STRING VARIABLE FOR CONVECTION BOUNDARY CONDITION
122 236

FI$ - 'FILEINFO.TXT'
1 178 178 178 182 182 182 186
198 198 198 200 200 200

FO - SURFACE VALUE OF B.C.
32 32 32

G - PARAMETER IN &D(G) SETS MIXED TEXT AND GRAPHICS
50 298

H - COUNTER OF SURFACE CONVECTION B.C.
32 32 32 38 144 144 144 144
146 146 316 318 328 330 330
330

HC - SURFACE CONDUCTANCE
14 126 126 126 126 126

I - GENERAL INDEX
12 12 12 12 12 12 30 44 50
50 50 54 54 54 54 58 58 58
58 58 58 66 66 66 66 78 78
78 78 78 78 78 78 94 94 94
94 98 98 98 98 98 98 98 98
98 106 106 106 106 106 106
138 138 138 138 138 138 138
138 144 144 144 144 144 144
156 156 156 156 158 158 158
166 166 166 166 166 166 212
212 224 224 224 232 262 262
262 262 266 266 266 274 274
274 280 280 280 280 280 280
280 280 294 294 294 294 300
300 300 300 302 302 302 304
304 304 304 304 304 304 304
304 304 304 304 304 312 312
312 312 312 312 316 316 316
316 316 318 318 318 318 318
318 322 322 322 322 322 322
322 322 330 330 330 334 334
336

I(*,*) - & INPUT CMD
304 306

I(*,*,*) - & INPUT CMD
4 72 78 82 102 126 128

I(*,*,*,*) - & INPUT CMD
40 88 124 134 136 208 256

I(*,*,*,*,*) - & INPUT CMD
66 76 90 114 116 120 122 126
132 180 268 276

I(*,*,*,*,*,*) - & INPUT CMD
70 88 94 188 244 248 294

I1 - STARTING SUBSCRIPT FOR A BOUNDARY
18 18 18 20 20 20 50 50 50
50 112 134 158 158 224 224
224 224 318 318

I2 - ENDING SUBSCRIPT FOR A BOUNDARY
18 18 20 20 50 50 50 50 112
134 158 158 224 224 224 224

IN - FLAG TO INDICATE INTERIOR NODE BOUNDARY CONDITION (1=YES; 0=NO)
8 8 8 8 8 18 18 124 124 130
130 222 222 222

IO - INPUT OPTION (1=REGION; 2=ELEMENT)
54 70 70 70 72 76 76 76 76
76 76 80 88 88 88 88 90 90
90 90 90 90 90

IO$(*) - INPUT OPTION DESCRIPTOR
70 70 76 88 88 90 232 232
232

IS(*) - ARRAY FOR START & END SUBSCRIPTS FOR BOUNDARIES
50 50 112 112 134 134 156 158 158

J - GENERAL SUBSCRIPT
12 12 12 12 12 12 12 12 12
12 12 12 12 12 32 32 32 32
138 138 138 138 280 280 280
280 300 302 302 322 322 322
322

J1 - GENERAL SUBSCRIPT
62 62 62 62 64 64 64 64 74
74 74 94 94 94 106 106

K - GENERAL SUBSCRIPT
22 22 22 22 24 24 24 24 36
36 36 38 38 38 46 46 46 46
46 48 48 48 48 48

KE - FLAG FOR ISOTROPIC THERMAL CONDUCTIVITY (1=YES; 0=NO)
66 66 74 84 92 94 104 106

KW$ - KEYWORD
192 192 192 198 198 228

L - LINE LENGTH
30 30 30 30 44 44 44 158 158
158

L$ - GENERAL LABEL STRING
54 62 62 62 64 64 86 86 100
106 106

L(*) - & LISTING CONTROL
1 286 290 310 312 316 316 318

LN% - LINE NODE NUMBER ARRAY
168 170

LN%(*,*) - LINE NODE NUMBER ARRAY
168 300 300

M(*) - & MATRIX CMD
66 228

MO - MENU OPTIONS
244 244 244 244 244 260 260
280 280 330

MP - MATERIAL PROPERTY ARRAY
66 92 92 256 258 260 264 268
290 296 300 302

MP$ - MATERIAL PROPERTY DEFAULT
80 80 82

MP(*,*) - MATERIAL PROPERTY ARRAY
66 66 66 72 72 74 74 74 74
74 74 80 84 84 84 92 92 92
92 92 92 106 106 106 256 258
262 266 286 286 288 288 298
302 302 306

N - SUBSCRIPT; COMMONLY IN STEPPING FUNCTION
8 8 8 8 10 10 14 14 18 18
18 18 18 18 18 18 18 18 18
20 20 20 20 20 20 20 20 20
22 22 22 22 22 22 24 24 24
24 26 26 26 30 30 30 30 34
34 34 34 42 50 50 50 50 60
114 154 154 154 158 158 158
158 158 158 158 222 222 222
222 222 224 224 224 224 224
224 224 224 224 224 312 312
312

N$ - GENERAL STRING
140 148 162 162 164 170 172
178 180 190 194 260 264 280
328

N0 - GENERAL INDEX
30 30 42 44 44 334

N1 - GENERAL INDEX; NUMBER OF ROWS IN SAVED ARRAY
20 20 22 22 24 24 26 26 30
30 30 32 32 32 32 36 36 36
36 36 36 36 36 36 50 50 50
54 54 54 138 146 154 154 154
158 158 160 162 162 162 168
170 178 182 182 196 258 262
278 300 300 300 326

N1% - NUMBER OF ITEMS FOUND IN A SEARCH
302

N2 - GENERAL INDEX; NUMBER OF COLUMNS IN SAVED ARRAY
20 22 22 24 24 30 30 30 32
32 32 36 36 36 36 36 54 54

54 138 146 158 158 160 178
182 182 196 258 262 300 300
300

N3 - NODE NUMBER
36 36 36 36 36

NA - GENERAL INDEX
10 10 10 10 10 12 14 30 30
32 32 34 34 34 36 36 38 42
42 42 42 42 60 60 60 60 60
60 144 144 144 146 146 146
146 146 146 224 224 224 224

NA$ - FILENAME
174 178 178 180 180 180 182
188 188 190 192 192 192 194
194 208

NB - # OF BOUNDARY NODES IN BN%
22 22 24 24 156 170 170

NC - # CODED PROPERTY VALUES
76 76 76 78 78 78 80 80 82
84 84 98 100 100 102 104 104
104 300 302 302 302 302 302
302 302 304 304

ND% - ELEMENT NODE #S
160 162 162 162

ND%(*,*) - ELEMENT NODE #S
36 36 36 36 54 54 100 106
154 154 160 162 224 224 224
226 226 226 226 226 226 334
334

NE - NUMBER OF ELEMENTS
26 28 28 66 68 74 84 90 90
90 92 92 92 100 106 154 160
160 256 262 262 266 274 274
286 288 300 302 302 306

NE$(*) - STRING FOR NODE OR ELEMENT LABEL
228 228 230 312

NI - NUMBER OF INPUT VALUES
10 10 12 12 12 12 12 14 14
14 14 14 14 14 14 60 108 138
138 138 138 144 144 146 146
278 278 280 312 322

NL - # UNIQUE LINES
168 168 300

NM - NEXT MENU DEFAULT
238 244

NN - NUMBER OF NODES
8 18 18 32 38 144 162 162
162 162 166 166 166 166 166
316 318 318 330 334 334 336

NO - GENERAL SUBSCRIPT
8 8 8 8 8 8 10 14 42 44 44
46 46 222 312 312 330 330 330
330 334

NP - NUMBER OF DATA POINTS
144 144 146 318 326 326

NQ - NUMBER OF QUADRILATERAL REGIONS
68 74 90 90 90 104 154 156
162 256

NS - # SEPARATE BOUNDARIES
50 134 134 154 156 158

NU - INDEX
44 44 56 58

NV - NORMAL VECTORS TO BOUNDARY
44 44 56 58 156

NV(*,*) - NORMAL VECTORS TO BOUNDARY
156 158 158

NX - X COMPONENT OF NORMAL VECTOR
30 32 44 44 44

NY - Y COMPONENT OF NORMAL VECTOR
30 32 44 44 44

O(*,*,*,*) - & ORDER CMD (SORT)
92 92 162 166 166 300

OP%(*) - OPTION CODE FOR &F()
54 228 302

P$ - STRING "NODE"
1 20 20

P% - LISTING SPEED PARAMETER
1 286 310 316

P2 - TWO * PI
30 224

PD$ - PROBLEM DESCRIPTOR
198 198

PN$ - PROGRAM NAME OPTIONS
250

PN$(*) - PROGRAM NAME OPTIONS
250 250 250 250 252 254

PV - COMMON VALUES OF MATERIAL PROPERTIES
296 298

PV(*) - COMMON VALUES OF MATERIAL PROPERTIES
298 302 302 302 304

R(*,*) - & RECALL ARRAY
162 162 164 170 172 260 280 328

R0 - REGION #
74 90 90 90 100

R1 - REGION # OR RECORD NUMBER
68 90 100 150 200 266

R2 - REGION # OR RECORD #
68 76 76 76 100 106 150 200 266

RE - REGION # OR RECORD #
54 54 74 74 74 74 74 90 90
90 90 100 100 100 100 100 100
100 102 104 104 104 104 104
104 106 106 106 106 138 146
154 154 154 154 154 154 154
156 156 156 156 156 156 156
156 160 160 162 162 168 168
170 170 178 178 182 182 182
200 200 200 200 258 258 262
272 272 278 278 326 326

RP - REGION PROPERTY
256 268

RP$ - REGION PROPERTY DEFAULT
100 100 102

RP(*,*) - REGIONAL MATERIAL PROPERTIES
74 74 74 100 104 104 104 106
256

S$ - 'STARTING'
1 20

S(*,*) - & SAVE ARRAY CMD
140 148 264

SB - SCREEN BOTTOM Y COORDINATE
152 232 232

SC - SCALE
152 152 152 152 152 232 232

SF$(*) - LABEL FOR CONVECTION BOUNDARY CONDITION (SIDE OR FACE OF ELEMENT)
230 230 230 318

SI%(*) - STATUS INDICATOR FOR RECORD(0 =UNDEFINED)
142 150 160 162 168 170 178
182 182 200 224 258 268 272
278 280 312 322 326 332

SL - SCREEN LEFT X COORDINATE
152 232 232

SR - SCREEN RIGHT X COORDINATE
152 232

SR$ - 'SELECT A NUMBER' PROMPT
1 70 88 94 244 248 294

ST - SCREEN TOP Y COORDINATE
152 232

SY - SYMBOL NUMBER
48 48 58 58

T - &D(T) SETS TEXT DISPLAY
86 106 110 136 212 238

T$ - STRING DEFINING ELEMENT TYPE
198 222

T1 - TYPE OF MATERIAL PROPERTY
66 66 66 68 70 94 94 94 94
94 94 262 294 296

T2 - TYPE OF MATERIAL PROPERTY
68 92 94 262 284 286 292 292
294 294 294 296

T3 - USED TO HANDLE VARIABLE MENU ITEMS IN PREPROCESS
292 294 294 294 294 294 294

TE - TYPE OF ELEMENT (1=TWO-DIMENSIONAL; 2=AXISYMMETRIC)
30 30 32 32 32 32 66 66 94
118 122 198 198 234 234 236
262 284 284 292 292

TE$ - DESCRIPTOR OF ELEMENT TYPES
222

TE$(*) - DESCRIPTOR OF ELEMENT TYPES
198 222 222 222

TF - TEMPERATURE OF FLUID FOR BOUNDARY CONDITION
126 126 126 126

TF$ - TEMPORARY FILE FOR SAVING ELEMENT THICKNESS
228 266 266 266 266 266 274
274 274

TH - THICKNESS OF ELEMENT
150 274

TH(*) - THICKNESS ARRAY FOR ELEMENTS
30 274 274

TY - TYPE OF MATERIAL PROPERTY (1=X-THERMAL CONDUCTIVITY; 2=Y-THERMAL CONDUCTIVITY; 3=THICKNESS)
68 68 68 70 70 70 70 70 70
72 72 72 72 74 74 74 74 74
74 76 76 78 78 78 80 82 84
84 84 88 92 92 92 92 92 92
92 92 92 92 94 94 94 98 100
102 104 104 104 106 106 106
286 286 286 286 286 286 288
288 288 288 290 290 290 294
294 294 294 294 294 294 294
294 294 296 296 296 296 296
296 298 298 298 300 302 302
306 306

TY$(*) - DESCRIPTOR OF MATERIAL PROPERTY TYPES
66 66 68 68 70 72 72 76 76
78 82 88 92 94 102 228 236
236 236 286 286 286 288 288
294 298 298 306

U$ - STRING DESCRIBING KEYS FOR SELECTING FOR LABELS
1 18 20 20 28

UH - HIGH VALUE OF X COORDINATE IN SPECIFIC PLOT
152 232

UL - LOW VALUE OF X COORDINATE IN SPECIFIC PLOT
152 232 232

UN - UNIFORMITY OF MATERIAL PROPERTY FLAG (0=NOT, 1=UNIFORM)
72 72 74 74

UN$(*) - UNIFORMITY DESCRIPTOR
92 230 230

VB - BOUNDARY VALUE
14 42 46 48 58 58 126 128
128 330

VH - HIGH VALUE OF Y COORDINATE IN SPECIFIC PLOT
152 232

VL - LOW VALUE OF Y COORDINATE IN SPECIFIC PLOT
152 232 232

W(*,*,*,*,*,*) - & WHICH CMD (SELECTS ROWS)
302

WR$ - 'WHEN READY' PROMPT
1 4

X - FUNCTION DUMMY ARGUMENT
232 232

X1 - GENERAL X COORDINATE; START OF LINE
26 26 26 30 30 32 32 44 44
44 54 54 56 56 58 158 158
232

X2 - GENERAL X COORDINATE; END OF LINE
30 30 32 32 44 44 44 54 54
56 158 158 232

XH - MAX X COORDINATE
166 232

XL - MIN X COORDINATE
166 232

XN - X COORDINATE OF NODE
162

XN(*) - X COORDINATE OF NODE
8 8 30 30 36 36 36 36 36 36
44 44 46 50 50 54 54 154 158
158 162 166 226 226 226 300
300

XP - GENERAL SCALED X COORDINATE
54 54 62 64 64 100

XY - COORDINATES OF POINTS
162 164 166 166 166

XY(*,*) - COORDINATES OF POINTS
162 166 166 166 166 166 166

Y - GENERAL Y COORDINATE
232 232

Y1 - GENERAL Y COORDINATE; STAR6 OF LINE
26 26 26 30 30 44 44 44 54
54 56 56 58 158 158 232

Y2 - GENERAL Y COORDINATE; END OF LINE
30 30 44 44 44 54 54 56 158
158 232

YH - MAX Y COORDINATE
166 232

YL - MINIMUM Y COORDINATE
166 232

YN - Y COORDINATE OF NODE
162

YN(*) - Y COORDINATES OF NODES
8 8 30 30 36 36 36 36 36 36
44 44 46 50 50 54 54 154 158
158 162 166 226 226 226 300
300

YP - GENERAL SCALED Y COORDINATE
54 54 62 64 64 100

Z - USER INTERACTION INDICATOR (0=MAX; 1=INTERMEDIATE; 2=FREE RUN; 3=DEMO)
6 94 136 180 180 188 188 196
222 238

1 FEB 85 Version JRC/DCD

END OF VAR. LIST

```
*******************************
*                             *
*         SOLVE.HEAT          *
*                             *
*     TABLE OF VARIABLES      *
*                             *
*******************************
```

*AR(*)
30 76 204

*P1(*) - STEPPING FUNCTION (CW) AROUND ELEMENT
30 206

*P2(*) - STEPPING FUNCTION (CCW) AROUND AN ELEMENT
30 72 76 206

A - GENERAL VARIABLE
76 76 76 76

A$ - GENERAL STRING VARIABLE; FREQUENTLY INPUT STRING
4 6 6 124 124 188 188 192 192

A2 - TWICE THE AREA OF AN ELEMENT
30 30 30 36 44

B - SHAPE FUNCTION DERIVATIVE MATRIX
30 40 44 44

B(*,*) - & BEEP COMMAND; DERIVATIVES OF SHAPE FUNCTIONS OF AN ELEMENT
4 6 30 30 38 40 198 222 228

B(*,*,*) - & BEEP (WITH REPETITION)
228 234

BC - EQUIVALENT BOUNDARY CONDITION MATRIX
206 208

BC% - BRANCH CODE (1 ='<' BACKWARD; 2 ='>' FORWARD BRANCH; 3 =':' A PREVIOUS VALUE
4 6 8 44 70 98 124 124 134
150 154 154 164 164 202 216
220 220 222 222 230 230

BC(*,*) - EQUIVALENT BOUNDARY CONDITIONS
18 18 52 52 72 72 74 76 76
206 208 208

BT - TYPE OF B.C. (0=INSULATED; 1=NODE HEAT SOURCE; 2=NODE TEMPERATURE; 3=SURFACE HEAT FLUX; 4=SURFACE CONVECTION); TRANSPOSE OF B MATRIX
40 44 44

BT(*,*) - TRANSPOSE OF B MATRIX
40

BV - BOUNDARY VALUE
18 18 20 22

BW - BANDWIDTH OF CONDUCTION (STIFFNESS) MATRIX
12 12 12 18 40 58 68 80 90
108 122 126 126 126 142 142
202 202

C - INTERMEDIATE MATRIX IN STIFFNESS MATRIX
40 44 44

C(*) - WEIGHTING FACTOR FOR ENDS OF CONTOUR LINES; CLEAR ARRAY (&C(*))
124 134 206

C(*,*) - & CLEAR ARRAY CMD; INTERMEDIATE MATRIX IN STIFFNESS MATRIX
40 198 204

C(*,*,*,*,*,*) - & CLEAR ARRAY CMD
202

C(*,*,*,*,*,*,*,*,*,*) - & CLEAR ARRAY CMD
40

CR$ - STRING FOR CARRIAGE RETURN
204

CR$(*) - STRING FOR CARRIAGE RETURN
38 140 142 158 172 204 204
204

D - ELEMENT CONDUCTIVITY MATRIX
26 40 44

D$ - CHR$(4) CONTROL D
1 152 152 152 152 156 156 156
162 162 174 174 174 178 178
178 188 192 226

D(*) - & DISPLAY (&D(T)=TEXT,&D(G)=MIXED,&D(V)=FULL GRAPHICS)
1

D(*,*) - ELEMENT MATERIAL PROPERTY (CONDUCTIVITY) MATRIX
26 26 26 26 40

D1$ - ROW DESCRIPTOR FOR SAVED DATA
152 156 158 172

D2$ - COLUMN DESCRIPTOR FOR SAVED DATA
152 156 158 172

DD - DIMENSION OF ELEMENT MATERIAL PROPERTY MATRIX
26 36 40 40 40 40 40 202

DE - # NODES / ELEMENT
28 30 36 40 40 40 76 76 202
202 202 206 206 206

DE$ - DESCRIPTION OF DATA IN FILEINFO.TXT
152 154 156 158 164 172

DF - DEGREE OF FREEDOM
18 18 18 18 18 18 18 18 22
22 22 22 22 22 24

DK O DIMENSION OF ELEMENT STIFFNESS MATRIX
38 38 40 40 40 40 40 46 46
46 48 48 202

DR - DISKETTE DRIVE NUMBER
152 156 162 166 174 178 188
192 198 198 222

DR$ - DISKETTE DRIVE STRING
198

DR$(*) - DISKETTE DRIVE STRING
152 156 162 166 174 178 188
192 198 198 198

E2 - ERROR FLAG
162 162 190

EF - ERROR FLAG (1=ERROR OCCURRED; 0=NONE)
56 58 60 62 68 68 82 84 90
90 94 94 106 108 110 112 120
120 122 122 130 132 136 136
146 146 148 148 180 206 206
208 208

EL - ELEMENT # (GENERAL)
8 26 26 28 30 36 40 40 42
42 42 42 44 46 52 72 72 72
74 76 76 76 76 204

EN% - NODE # OF ELEMENT
146 202

EN%(*,*) - ELEMENT NODE #S (ELEMENT#, LOCAL NODE #)
28 72 72 76 76 202 202

EO - EXIT OPTION (1=PROCEED TO NEXT PROGRAM; 2=TO MAIN MENU; 3=REMAIN IN SAME; 0=STOP)
210 220 220 220 220 220 222
224 226

ER - ERROR CODE
180 180 180 180 180 180 180
180 180 180 180 180 180 180
188 188 188 188 188 188 188

ER$ - ERROR CODE STRING DESCRIPTORS
198

ER$(*) - ERROR CODE STRING DESCRIPTORS
188 198 198 200 200 200 200
200

FI$ - 'FILEINFO.TXT'
1 152 152 152 156 156 156 162
174 174 174 178 178 178

FR - FREE RUN FLAG (1=FREE RUN; 0=USER CONTROL)
4 4 4 4 10 40 70 96 124 134
154 158 158 164 172 172 210
210 222 230 230 230 230 230
234

G1 - PIVOT VALUE FOR GAUSSIAN ELIMINATION
12 14 14

G2 - STIFFNESS VALUE
14 14 14

GF - GLOBAL FORCE (SOURCE) MATRIX
40 56 94 106 120 202

GF(*) - GLOBAL FORCE (SOURCE) VARIABLE
14 14 14 18 18 20 20 22 40
52 126 126 140 202

GS - GLOBAL STIFFNESS MATRIX
40 60 68 82 90 110 122 202

GS(*,*) - GLOBAL STIFFNESS MATRIX
12 14 14 14 14 18 18 20 20
22 22 22 22 22 40 48 48 48
74 74 74 74 74 74 76 76 76
76 76 76 126 126 126 142 202

H - HORIZONTAL INDEX
32 32 32 32 32 32 40 124

H1 - HORIZONTAL INDEX
32 32 32 32 32

HA - PRODUCT OF CONVECTION COEFFICIENT AND AREA
74 74 74 74

HB - NUMBER OF CONVECTION BOUNDARY CONDITIONS ON ELEMENT SIDES
72 72 208

HF - NUMBER OF CONVECTION BOUNDARY CONDITIONS ON FACES
76 208

I - GENERAL INDEX
12 12 12 12 12 12 14 14 14
14 14 26 26 26 26 26 30 30
30 34 34 34 34 36 36 36 36
36 36 38 38 38 46 46 46 46
46 46 46 46 48 48 48 50 52
52 52 52 52 72 72 72 74 74
76 76 76 76 126 126 126 126

126 126 126 126 126 126 126
126 126 138 138 138 138 140
140 140 140 142 142 142 142
194 194 206 206 206 206 206 206

I(*,*,*) - & INPUT CMD
4

I(*,*,*,*) - & INPUT CMD
6 188 192

I(*,*,*,*,*) - & INPUT CMD
124

I(*,*,*,*,*,*) - & INPUT CMD
8 70 98 154 164 216 220 230

J - GENERAL SUBSCRIPT
14 14 14 14 14 14 18 18 18
18 20 20 22 30 30 30 30 30
30 38 38 38 38 38 48 48 48
50 74 74 74 76 76 76 76 126
126 126 126 142 142 142 142

K - GENERAL SUBSCRIPT
14 14 14 14 14 18 20 20 20
20 20 20 20 28 28 28 28 28
28 28 28 30 30 30 46 46 46
46 48 48 48 48 74 74 74 76
76 76 126 126 126

KE - ELEMENT STIFFNESS MATRIX
40 44 44 44

KE(*,*) - ELEMENT STIFFNESS MATRIX
40 46 48

KI - GENERAL INDEX
48 48 48 48 48 48 48

KJ - GENERAL INDEX
48 48 48 48

KW$ - KEYWORD
168 168 168 174 176 176

L - LINE LENGTH; SUBSCRIPT
72 74 76 76 76

L(*) - & LISTING CONTROL
1 26 26 34 34 36 38 44 46
48 48 138 138 140 140 142 142

LA$ - GENERAL LABEL
72 72 78 78 98 100 102

M - SUBSCRIPT
18 18 18 18

M(*) - & MATRIX CMD
26 30 44 44 44 44

MJ - SUBSCRIPT
12 12 12 12 12 14 126 126

MK - SUBSCRIPT
12 12 14 14 14

ML - LABEL FLAG FOR MATRIX PRINTING
72 78 98 98 100 102

MO - MENU OPTIONS
6 210 216 216 216 218 218

MP - MATERIAL PROPERTY MATRIX
202 206

MP(*,*) - MATERIAL PROPERTY MATRIX
26 26 44 74 202 206 206 206

N - SUBSCRIPT
12 14 14 14 14 126 126

N$ - GENERAL STRING
56 60 68 82 90 94 106 110
120 122 130 136 146 148 166
206 208

N1 - GENERAL INDEX; NUMBER OF ROWS IN SAVED ARRAY
54 58 68 72 72 72 74 74 74
74 74 74 76 76 76 76 76 76
76 80 90 94 104 108 120 122
128 134 136 152 152 156 158
172 202 202 206

N2 - GENERAL INDEX; NUMBER OF COLUMNS IN SAVED ARRAY
54 58 68 72 72 72 74 74 74
74 74 74 76 76 76 76 76 76
76 80 90 104 108 122 128 152
156 158 172 202

NA$ - FILENAME
150 152 154 154 156 158 164
164 166 168 168 168 170 172
188

NC - NODE COORDINATES
148 202

NC(*,*) - COORDINATES OF NODES (ABSOLUTE OR SCREEN COORDINATES)
28 28 34 34 72 72 72 72 202

ND - NODE #
40

ND(*) - NODE NUMBERS FOR AN ELEMENT
28 28 28 36 40 46 46 48 48

NE - NUMBER OF ELEMENTS
42 42 202 202 202

NM - NEXT MENU DEFAULT
210 216

NN - NUMBER OF NODES
12 12 12 12 12 12 12 18 18
18 40 40 52 54 58 68 72 72
72 80 90 94 104 108 120 122
124 126 126 126 126 126 126
128 136 138 140 140 142 202
202 202 202

NO - GENERAL SUBSCRIPT
72 72 76

NP - NUMBER OF DATA POINTS
72 76 206 206

P% - LISTING SPEED PARAMETER
1 34 138 140 142

PD$ - PROBLEM DESCRIPTOR
174 176

PI - CONSTANT PI
44 74 206

PN$ - PROGRAM NAME OPTIONS
204

PN$(*) - PROGRAM NAME OPTIONS
204 204 204 222 226

PO - PRINT OR PLOT OPTION
8 8 8 8 26 30 40 42 42 44
46 48 48 50 50 50 50 50 50
52 70 70 70 70 72 72 78 78
96 98 98 98 98 98 98 98 102
102 102 102 102 102 124 124
126 126 126 126

R(*,*) - & RECALL ARRAY
68 90 94 120 122 136 146 148
206 208

R1 - RECORD NUMBER
62 84 112 132 178

R2 - RECORD #
62 84 112 132 178

RB - EFFECTIVE RADIUS
28 44

RE - RECORD #
54 58 64 64 64 80 86 86 86
92 92 92 104 104 108 108 114
114 114 120 120 128 134 134
134 144 144 146 146 152 152
156 156 156 178 178 178 178
202 202 204 206 218 218 218

S - ELEMENT SIDE #
72 72 72

S(*,*) - & SAVE ARRAY CMD
56 60 82 106 110 130

SI% - FILE STATUS INDICATOR
202

SI%(*) - STATUS INDICATOR FOR DATA FILE(0 =UNDEFINED)
64 64 86 86 92 92 104 108
114 114 120 134 134 144 146
152 156 156 178 202 218

SR$ - 'SELECT A NUMBER' PROMPT
1 8 70 98 216 220 230

SU - SUM IN MATRIX MULTIPLICATION
126 126 126 126

T - &D(T) SETS TEXT DISPLAY
1

TE - TYPE OF ELEMENT (1=TWO-DIMENSIONAL; 2=AXISYMMETRIC)
44 44 74 74 74 74 74 74 74
74 174

TRN(*) - TRANSPOSE FUNCTION
44

U$ - UNDERLINE STRING
138 200 200

UN - UNIFORMITY OF MATERIAL PROPERTY FLAG (0=NOT, 1=UNIFORM)
42 44 206

UU - TEMPERATURES OF NODES
124 130 134 136

UU(*) - TEMPERATURES OF NODES
34 124 126 126 126 134 138

V - VERTICAL INDEX
32 32 32 32 32 32 40 124

VE - VOLUME OF ELEMENT
44 44

WR$ - 'WHEN READY' PROMPT
1 4

X - X COORDINATE MATRIX
40

X(*) - X COORDINATES OF NODES OF AN ELEMENT
28 28 28 28 28 28 28 28 28
28 30 30 36 40 204 204 204
204 204 204

X1 - GENERAL X COORDINATE; START OF LINE
72 72 74 74 74

X2 - GENERAL X COORDINATE; END OF LINE
72 72 74 74 74

Y - Y COORDINATE MATRIX
40

Y(*) - Y COORDINATES OF NODES OF AN ELEMENT
28 30 30 36 40 204 204 204
204 204 204

Z - USER INTERACTION INDICATOR (0=MAX; 1=INTERMEDIATE; 2=FREE RUN; 3=DEMO)
1 154 154 158 164 164 172 210
210 210 216 230

1 FEB 85 Version JRC/DCD

END OF VAR. LIST

```
*******************************
*                             *
*      POSTPROCESS.HEAT       *
*                             *
*     TABLE OF VARIABLES      *
*                             *
*******************************
```

*AN(*) - ANGLE CCW FROM X TO GRADIENT VECTOR
36 64 172

*LL(*) - DEFINES LENGTH OF LINE
104 172

*P1(*) - STEPPING FUNCTION (CW) AROUND ELEMENT
16 170

*P2(*) - STEPPING FUNCTION (CCW) AROUND AN ELEMENT
16 170

A - GENERAL VARIABLE
32 58

A$ - GENERAL STRING VARIABLE; FREQUENTLY INPUT STRING
4 20 20 50 50 150 150 154
154

A2 - TWICE THE AREA OF AN ELEMENT
14 16 16 52

B - BOUNDARY NUMBER; SHAPE FUNCTION DERIVATIVE MATRIX
16 30 102 102 102 104

B(*,*) - & BEEP COMMAND; DERIVATIVES OF SHAPE FUNCTIONS OF AN ELEMENT
4 16 16 30 36 162 186

B(*,*,*) - &BEEP (WITH REPETITION)
190 198

BC% - BRANCH CODE (1 ='<' BACKWARD; 2 ='>' FORWARD BRANCH; 3 =':' A PREVIOUS VALUE
4 20 20 32 44 50 50 58 70
70 120 120 130 130 182 182
184 184 184 186 186 196 196 198 200

BF - BOUNDARY HEAT FLUX
98 108 110

BF(*,*) - BOUNDARY HEAT FLUX
98 104 104 104

BN% - BOUNDARY NODES
98 100 110

BN%(*,*) - BOUNDARY NODES ARRAY WITH NODES IN ORDER AROUND BOUNDARIES
98 100 100 104 104 104

BW - BANDWIDTH OF CONDUCTION (STIFFNESS) MATRIX
6 6 6 48 62 62 62 82 82 88
166

C(*) - CLEAR ARRAY (&C(*))
42 58 60 68 72 74 76

C(*,*) - & CLEAR ARRAY CMD
168

C(*,*,*) - & CLEAR ARRAY (3 PARMW
18 82 96 164 166

C(*,*,*,*) - & CLEAR CMD
30 62 98 110

C(*,*,*,*,*) - & CLEAR ARRAY (5 PARM)
48

C(*,*,*,*,*,*) - & CLEAR ARRAY CMD
162

C1 - GENERAL VARIABLE
6 8 52 52 52 52

C2 - GENERAL VARIABLE

8 8 8 52 52

CE - ELEMENT CONJUGATE STIFFNESS MATRIX
48 52 62

CE(*,*) - ELEMENT CONJUGATE STIFFNESS MATRIX
48 52 54 56

CR$ - STRING FOR CARRIAGE RETURN
164

CR$(*) - STRING FOR CARRIAGE RETURN
124 138 164 168 168

CS - CONJUGATE STIFFNESS MATRIX
48 62

CS(*,*) - CONJUGATE STIFFNESS FOR NODE TEMP GRADIENT CALCULATIONS
6 8 8 8 8 48 56 56 56 62
62 62

D$ - CHR$(4) CONTROL D
1 118 118 118 122 122 122 128
128 140 140 140 150 154 188

D(*) - & DISPLAY (&D(T)=TEXT,&D(G)=MIXED,&D(V)=FULL GRAPHICS)
1

D1$ - ROW DESCRIPTOR FOR SAVED DATA
118 122 124 138

D2$ - COLUMN DESCRIPTOR FOR SAVED DATA
118 122 124 138

DD - DIMENSION OF ELEMENT MATERIAL PROPERTY INDEX
8 30 30 36 36 36 38 48 48
52 56 60 62 64 64 64 98 166

DE - # NODES / ELEMENT
14 16 22 22 34 48 48 48 52
54 54 54 56 56 166 166 166
170 170 170

DE$ - DESCRIPTION OF DATA IN FILEINFO.TXT
118 120 122 124 130 138

DF$ - COORDINATE (DIRECTION) DESCRIPTOR
164

DF$(*) - COORDINATE (DIRECTION) DESCRIPTOR
164 168 168 168 168 168 168

DI$ - GENERAL COORDINATE (DIRECTION) DESCRIPTOR
164

DI$(*) - GENERAL COORDINATE (DIRECTION) DESCRIPTOR
44 44 164 168 168 172 172

DK - DIMENSION OF ELEMENT STIFFNESS MATRIX
30 30 36 166

DR - DISKETTE DRIVE NUMBER
118 122 128 134 140 150 154
162 162 186

DR$ - DISKETTE DRIVE STRING
162

DR$(*) - DISKETTE DRIVE STRING
118 122 128 134 140 150 154
162 162 162

E2 - ERROR FLAG
128 128 152

EF - ERROR FLAG (1=ERROR OCCURRED; 0=NONE)
20 20 26 28 32 32 40 42 50
50 66 68 76 76 84 84 86 86
94 96 100 100 100 100 108 110
114 114 116 116 144 194 194

EG - TEMPERATURE GRADIENT MATRIX FOR ELEMENTS
48 50 58

EG(*,*) - TEMPERATURE GRADIENT MATRIX FOR ELEMENTS
48 52

EL - ELEMENT # (GENERAL)
12 12 14 22 22 22 22 22 22
22 22 34 34 36 36 36 36 36
36 36 36 52 52 52 54 54 56
56 58 104 104 104 104 104

EN% - NODE # OF ELEMENT
114 162

EN%(*,*) - ELEMENT NODE #S (ELEMENT#, LOCAL NODE #)
14 22 54 54 56 56 166 166

EO - EXIT OPTION (1=PROCEED TO NEXT PROGRAM; 2=TO MAIN MENU; 3=REMAIN IN SAME; 0=STOP)
176 184 184 184 184 184 186
188

ER - ERROR CODE
144 144 144 144 144 144 144
144 144 144 144 144 144 144
150 150 150 150 150 150 150
152

ER$ - ERROR CODE STRING DESCRIPTORS
162

ER$(*) - ERROR CODE STRING DESCRIPTORS
150 162 164 164 164 164 164
164

FI$ - 'FILEINFO.TXT'
1 118 118 118 122 122 122 128
140 140 140

FR - FREE RUN FLAG (1=FREE RUN; 0=USER CONTROL)
4 4 4 4 20 32 50 58 58 76
90 92 120 124 124 130 138 138
176 182 186 196 196 196 196
196 198 198

N1 - GENERAL INDEX; NUMBER OF ROWS IN SAVED ARRAY
24 38 48 64 72 74 76 78 80
82 92 98 104 104 104 104 106
118 118 122 124 138 166 166

N2 - GENERAL INDEX; NUMBER OF COLUMNS IN SAVED ARRAY
24 38 48 64 72 74 82 92 104
104 104 104 106 118 122 124
138 166

NA$ - FILENAME
118 120 120 122 124 130 130
132 132 132 134 136 138 150
194 200

NB - # OF BOUNDARY NODES IN BN%
98 98 98 106

NC - NODE COORDINATES
116 162

NC(*,*) - ACTUAL COORDINATES OF NODES
14 14 104 104 104 104 166

ND(*) - NODE NUMBERS FOR AN ELEMENT
14 14 14 34

NE - NUMBER OF ELEMENTS
18 22 24 30 32 34 34 38 52
52 98 166 166 166

NM - NEXT MENU DEFAULT
176 182

NN - NUMBER OF NODES
6 6 6 6 6 6 6 18 30 48 48
58 60 62 62 62 62 62 62 62
62 64 64 82 82 82 82 86 86
88 90 92 166 166

NR - # ROWS IN A MATRIX
32 46 58 76

NS - # SEPARATE BOUNDARIES
100 100 102

P% - LISTING SPEED PARAMETER
1 22 46 54 78 80 90

PD$ - PROBLEM DESCRIPTOR
140 142

PN$ - PROGRAM NAME OPTIONS
166

PN$(*) - PROGRAM NAME OPTIONS
166 168 168 186 188

PO - PRINT OR PLOT OPTION
20 20 22 32 36 44 44 44 46
46 46 50 50 52 56 58 64

R(*,*) - & RECALL ARRAY
20 32 50 76 84 86 100 100
114 116 194

RD - RADIANS/DEGREE CONVERSION
170 172 172

RE - RECORD #; RIGHT SIDE ELEMENT MATRIX FOR CONJUGATE STIFFNESS EQUATION
18 18 24 30 30 38 48 48 48
52 52 62 64 72 72 82 82 84
84 92 98 98 100 100 106 112
112 114 114 118 118 122 122
122 166 166 192 202

RE(*,*) - RIGHT SIDE OF ELEMENT EQUATION FOR CONJUGATE SOLUTION
48 52 56

RH - RIGHT OF GLOBAL CONJUGATE EQUATION
48 62 82 94 96

RH(*) - RIGHT SIDE OF GLOBAL CONJUGATE EQUATION
82 90 90

RH(*,*) - RIGHT SIDE OF GLOBAL CONJUGATE EQUATION
8 8 8 48 56 56 62 62

RW - MATRIX ROW TYPE
32 46 58 76 78 88 88 88 88
88 88 88 88

RW$ - MATRIX OF ROW DESCRIPTORS
166

RW$(*) - DESCRIPTOR OF MATRIX ROW TYPE,RW
46 78 168 170 170

S(*,*) - & SAVE ARRAY CMD
26 40 66 94 108

SI% - STATUS INDICATOR (0=INACTIVE; 1=ACTIVE)
162

SI%(*) - STATUS INDICATOR FOR RECORD(0 =UNDEFINED)
18 30 48 58 72 82 84 98 100
112 114 118 122 122 162 202

SR$ - 'SELECT A NUMBER' PROMPT
1 44 70 182 184 196

SU - SUM IN MATRIX MULTIPLICATION
62 62 62 62 88 88 88 88 88
90

T - &D(T) SETS TEXT DISPLAY
1

TE - TYPE OF ELEMENT (1=TWO-DIMENSIONAL; 2=AXISYMMETRIC); ELEMENT TEMPERATURE MATRIX
18 26 82 86 96 140 168 168

TE(*) - TEMPERATURES OF ELEMENTS
12 18 22 22 22 22 22 82 88
88 90

TG - TEMPERATURE GRADIENT MATRIX OR OTHERS FOR LISTING
30 40 42 60 66 68 72 74 76
76 98 100 110

TG(*) - MATRIX FOR LISTING
74 78

TG(*,*) - TEMPERATURE GRADIENTS OR OTHER VARIABLES FOR LISTING
30 36 36 36 36 36 36 36 46
46 60 62 62 62 64 64 64 64
72 80 80 80 98 104 104

U - TEMPERATURES FOR NODES OF AN ELEMENT
18 30

U(*,*) - TEMPERATURES OF NODES OF AN ELEMENT
30 34 36

UN - UNIFORMITY OF MATERIAL PROPERTY FLAG (0=NOT, 1=UNIFORM)
170

UU - TEMPERATURE OF NODES FOR ALL NODES
18 20 30 32

UU(*) - TEMPERATURES OF NODES FOR ALL NODES
18 22 30 34

VA - GENERAL VARIABLE
72 78

VA$ - DESCRIPTOR OF VA
166

VA$(*) - DESCRIPTOR FOR VA
46 78 166 170 170 170

W - INDEX
86 88 88 88 90

X(*) - X COORDINATES OF NODES OF AN ELEMENT
14 14 14 14 14 14 14 16 16

X1 - GENERAL X COORDINATE; START OF LINE
36 36 36 64 64 64 104 104
104 172 172 172 172 172 172
172 172 172

X2 - GENERAL X COORDINATE; END OF LINE
104 104 172

Y(*) - Y COORDINATES OF NODES OF AN ELEMENT
14 14 14 14 14 14 14 16 16

Y1 - GENERAL Y COORDINATE; START OF LINE
36 36 64 64 104 104 172 172
172 172 172

Y2 - GENERAL Y COORDINATE; END OF LINE
104 104 172

Z - USER INTERACTION INDICATOR (0, MAX; 1,INTERMEDIATE; 2,FREE RUN; 3,DEMO)
1 120 120 124 130 130 138 176
182 182 182

Z5 - GENERAL VARIABLE
86 88 88 88

1 FEB 85 Version JRC/DCD

END OF VAR. LIST

```
******************************
*                            *
*         PLOT.HEAT          *
*                            *
*     TABLE OF VARIABLES     *
*                            *
******************************
```

*AO(*) - TEST FOR LINE ENTIRELY OFF SCREEN (1=OFF; 2=NOT)
26 28 30 268

*C(*) - STEPPING FUNCTION AROUND CONTOUR LINES
16 268

*LE(*) - DEFINES LENGTH OF LINE
36 60 68 268

*M(*) - STEPPING FUNCTION AROUND BOUNDARIES
108 270

*P2(*) - STEPPING FUNCTION (CCW) AROUND AN ELEMENT
38 84 270

*PX(*) - CONVERTS X ABSOLUTE COORDINATE TO SCREEN COORDINATE
60 60 66 114 116 132 132 274

*PY(*) - CONVERTS Y ABSOLUTE COORDINATE TO SCREEN COORDINATE
60 60 66 116 132 132 274

*S(*) - STEPPING FUNCTION AROUND A MATRIX OF NP VALUES (UP IF A=21; DOWN IF A=8)
12 270

*UX(*) - CONVERTS X SCREEN COORDINATE TO ABSOLUTE COORDINATE

112 114 274

*VY(*) - CONVERTS Y SCREEN COORDINATE TO ABSOLUTE COORDINATE
112 114 274

*W(*) - TEST FOR POINT OFF SCREEN (1=OFF;0=NOT)
40 58 64 72 76 270

*WM(*) - STEPPING FUNCTION AROUND ZONES BEING SELECTED FOR PLOTTING
124 124 280

*ZX(*) - DEFINITION OF X COORDINATE FOR DOT IN ZONE SELECTION
110 110 124 124 280

*ZY(*)
110 110 124 124 280

A - GENERAL VARIABLE; FREQUENTLY ASCII VALUE OF INPUT CHARACTER
12 12 12 12 12 12 12 16 16
16 18 18 18 18 18 18 18 40
50 58 64 72 76 96 108 118
118 118 124 196 196 196 196
196 202 202 202 202 210 210
210 268 268 270 270 270 270
270 280 280 280 280 304 314
314 314 316 316 316 318 318
318 318 318 322 324 326 326
326 326 326

A$ - GENERAL STRING VARIABLE; FREQUENTLY INPUT STRING
4 80 86 86 102 102 118 120
154 154 158 158 162 162 164
164 174 176 176 176 176 180
180 182 184 186 186 188 188
190 190 192 192 198 198 198
200 200 200 200 246 246 250
250

A% - GENERAL VARIABLE
84 86 166 178 182 186

A%(*) - FLAG USED IN LABELING POINTS ON PLOTS
12 18 18 18 48 166 168 176 182

AN - INPUT ANGLE (DEGREES); INCLUDED ANGLE
72 72 72 304 304

AN(*) - ANGLE, DEGREES
72 304 304

AS% - ASCII VALUE OF INPUT CHARACTER
6 6 6 6 6 6 6 44 44 44 44
48 48 48 48 48 48 134 134

B - BOUNDARY NUMBER
108 108 108 108 108 270 270
270 270 270 270 270 270 270

B(*,*) - & BEEP COMMAND
4 6 12 16 260 288 308

BC% - BRANCH CODE (1 ='<' BACKWARD; 2 ='>' FORWARD BRANCH; 3 =':' A PREVIOUS VALUE
4 104 104 104 104 104 112 118
118 120 120 120 144 146 150
150 152 152 152 154 156 156
166 166 180 180 180 182 188
188 188 188 196 196 196 198
198 220 228 228 272 284 284
286 286 288 288 292 294 294
298 298 302 304 306 306 308
310 314 318 318

BD - CODE FOR BOUNDARY CONDITION DIRECTION (1=X-DIRECTION;2=Y-DIR;3=NORMAL TO BOUNDARY (+OUT);4=NORMAL TO FACE (SCREEN))
56 56 66 66 66 66

BI - INPUT BOUNDARY CONDITION MATRIX
90 92 292

BI(*,*) - COMBINED BI% AND BV (INPUT BOUNDARY CONDITIONS)
56 56 56 56 64 90

BN% - BOUNDARY NODES
216

BN%(*,*) - BOUNDARY NODES ARRAY WITH NODES IN ORDER AROUND BOUNDARIES
56 56 108 108 272 272 274

BR% - BEGINNING ROW NUMBER IN &W()
86 322 322 322 322 322 326
326 326 326 326

BT - TYPE OF B.C. (0=INSULATED; 1=NODE HEAT SOURCE) 2=NODE TEMPERATURE; 3=SURFACE HEAT FLUX; 4=SURFAC E CONVECTION)
56 56 56 58 58 58 60 64

C - GENERAL
38 38 38 40 40 40 40 48 48 48 48

C% - CODE FOR COLOR OF FILL
76 76

C(*) - WEIGHTING FACTOR FOR ENDS OF CONTOUR LINES; CLEAR ARRAY (&C(*))
26 26 28 30 38 38 38 86 90
94 98 120 258 272 292 304 306
308

C(*,*) - & CLEAR ARRAY CMD
84 182 186 218 284 294 304 304

C(*,*,*,*) - & CLEAR CMD
166 178 208

CC - MATRIX OF ELEMENT CENTROID COORDINATES
204 204 204 208 218 218 218
330

CC(*,*) - ABSOLUTE COORDINATES OF ELEMENT CENTROIDS
204 204 204 208 218 218 218
218

CF - FLAG INDICATING CONTOUR POINT SAVED
48 50 164 180

CL - CONTOUR LINE INDEX; COORDINATES OF POINTS ON CONTOUR LINE
14 14 14 14 16 16 16 84 84
84 84 84 86 86 186 186 186

CL(*,*) - COORDINATES OF POINTS ON CONTOUR LINE IN AN ELEMENT
20 20 20 84

CO - CONTOUR LABEL OPTION (1=LETTER; 2=NUMBER)
16 180 180 180 180 184 186

CO(*) - CODE FOR COLOR IN &FILL
76 262

CT% - ASCII VALUE OF INPUT CHARACTER
6 44 44 44 44 44 44 44 44
48 48 48 48 48 48 48 48

CV - CONTOUR VALUE; COORDINATES OF CONTOUR LINE POINTS TO BE LABELED
16 16 20 84 84 84 84 86 186
186

CV(*,*) - COORDINATES OF CONTOUR LINE POINTS TO BE LABELED
14 14 14 16 16 16 16 16 16
16 50 50 84 84 88 88 88 88
186 186 186

D - GENERAL VARIABLE
48 48 48 48

D$ - CHR$(4) CONTROL D
1 200 222 222 222 222 226 236
236 236 246 250 262 262 290
310 312 316 332 332

D%(*,*) - LOGICAL VALUES INDICATING LOCATION OF LINE ENDS RELATIVE TO SCREEN
26 26 26 26 28 28 28 28 28
28 28 28 30 30 30 30 30 30
30 30 268 268 268 268 268 268
268 268

D(*) - & DISPLAY (&D(T)=TEXT,&D(G)=MIXED,&D(V) =FULL GRAPHICS)
44 44 52 54 112 114 118 134
166 174 176 178 182 184 186
186 190 194 254 282 292 304
306 318 320 328 330

D1$ - ROW DESCRIPTOR FOR SAVED DATA
222 234

D2$ - COLUMN DESCRIPTOR FOR SAVED DATA
222 234

DD - DIMENSION OF DIRECTION MATRIX
262 262 304

DE - # NODES / ELEMENT
84 204 204 218 218 270
270 272 272 272 272

DE$ - DESCRIPTION OF DATA IN FILEINFO.TXT
222 228 234

DF$(*) - COORDINATE (DIRECTION) DESCRIPTOR
262 268

DI$(*) - GENERAL COORDINATE (DIRECTION) DESCRIPTOR
262 268 278 278

DR - DISKETTE DRIVE NUMBER
198 222 230 236 246 250 258
258 288 308 310 312 316

DR$ - DISKETTE DRIVE STRING
258

DR$(*) - DISKETTE DRIVE STRING
222 230 236 246 250 258 258
258 258 312 316

DV - INCREMENT IN X OR Y VARIABLE; INCREMENT IN PLOT VARIABLE
16 80 80 84 102 102 104 104
104 104 104 184 186

DX - X DISTANCE
68 68 68 68 68 72 72 72

DY - Y DISTANCE
68 68 68 68 68 72 72 72

E2 - ERROR FLAG
226 226 248

EC - MATRIX OF ELEMENT CENTROID SCREEN COORDINATES
208 218 322 322 322 324 330

EC(*,*) - SCREEN COORDINATES OF ELEMENT CENTROIDS
64 64 64 70 70 76 76 204 208
218 218 322 322 322 330 330
330 330

EF - ERROR FLAG (1=ERROR OCCURRED; 0=NONE)
92 92 96 96 100 100 208 208
210 210 214 214 216 216 240
316 316

EL - ELEMENT # (GENERAL)
38 38 64 64 64 64 76 76 76
76 76 78 84 84 84 86 218 218
218 218 218 218 218 218 218

EN - INDEX FOR END OF LINE BEING CLIPPED OR PLOTTED
26 26 26 26 26 26 26 26 26
26 28 28 28 28 28 28 28 28
28 28 28 28 28 28 30 30 30
30 30 30 30 30 30 30 30 30
30 32 32 32

EO - EXIT OPTION (1=PROCEED TO NEXT PROGRAM; 2=TO MAIN MENU; 3=REMAIN IN SAME; 0=STOP)
286 286 286 286 286

ER - ERROR CODE
240 240 240 240 240 240 240
240 240 240 240 240 240 240
246 246 246 246 246 246 246

ER$(*) - ERROR CODE STRING DESCRIPTORS
246 262 264 264 264 264 264
264

ER% - ENDING ROW NUMBER RETURNED FROM A VARIABLE SEARCH
86 86 322 322 322 322 326 326
326 326

EX(*) - X SCREEN COORDINATE OF LINE END
14 14 20 22 22 26 26 28 28
28 28 30 30 30 30 30 30 34
34 36 36 38 38 38 60 60 64
64 64 66 66 66 68 68 72 72
106 108 108 262

EY(*) - Y SCREEN COORDINATE OF LINE END
14 14 20 22 22 26 26 28 28
28 28 28 28 30 30 30 30 34
34 36 36 38 38 38 60 60 64
64 64 64 66 66 66 68 68 72
72 106 108 108 262

F(*,*,*,*) - & FILL CMD
76

FI$ - 'FILEINFO.TXT'
1 222 222 222 236 236 236

FR - FREE RUN FLAG (1=FREE RUN; 0=USER CONTROL)
20 20

G - PARAMETER IN &D(G) SETS MIXED TEXT AND GRAPHICS
44 52 114 178 186 318 328

G$ - GENERAL INPUT CHARACTER
6 6 12 12 16 16 46 46 46 46
112 112 114 114 114 116 118
124 124 124 124 124 124 124
126

GC - GRADIENT COMPONENT OPTION
78 184 294 294 294 294 296
296 296 304 304

GC$(*) - DESCRIPTOR FOR GRADIENT COMPONENTS
78 184 262 278 278 280 294
296

GD - GRADIENT DIRECTION INDICATOR
160 160 160 160 160 160 292
296 298 302 302 302 302 302
306 306

H% - HORIZONTAL INDEX
154 154 154

HE$ - GENERAL HEADING
284 284 284

I - GENERAL INDEX
26 28 30 36 60 68 70 70 70
70 70 70 72 72 72 74 80 80
80 80 84 84 84 86 106 106
106 106 106 110 110 110 120
120 120 122 122 122 122 124
124 124 124 124 124 124 124
124 124 126 126 126 126 128
128 128 128 128 128 130 130
130 132 132 132 132 132 132
132 132 168 168 168 168 168
168 176 176 176 176 176 176
176 182 182 182 184 184 184
184 204 204 204 204 204 204
204 204 204 210 210 210 210
210 210 218 218 218 254 254

268 268 270 270 270 274 274
274 274 280 280 280 280 280
280 280 282 282 282 282 294
294 294 294 298 298 298 298
304 304 304 304 304 304 304
304 304 304 312 312 312 312
322 322 322 322 322 322 326
326 326 326 326 326 328 328
328 328 328 328 330 330 330
330 330 330

I% - LABELING STATUS INDICATOR
166 178 182 186

I%(*) - LABELING STATUS INDICATOR
12 18 18 48 48 48 166 168
176 176 182 182 186 322 326

I(*,*) - & INPUT CMD
80 174 182

I(*,*,*) - & INPUT CMD
4 184

I(*,*,*,*) - & INPUT CMD
154 158 162 164 176 176 180
186 188 188 190 192 246 250

I(*,*,*,*,*) - & INPUT CMD
86 102 104 104 104 118 120
180 228

I(*,*,*,*,*,*) - & INPUT CMD
150 166 196 198 284 286 294
298 302 314 318

I0
70 70 70

I1 - GENERAL SUBSCRIPT; STARTING SUBSCRIPT FOR A BOUNDARY
8 8 8 8 10 10 10 10 38 38
38 108 108 108 110 110 110
110

I2 - GENERAL SUBSCRIPT; ENDING SUBSCRIPT FOR A BOUNDARY
38 38 38 108 108 108

IS - ARRAY FOR START; END SUBSCRIPTS
272

IS(*) - ARRAY FOR START & END SUBSCRIPTS FOR BOUNDARIES

108 108 270 270 270 270 270
270 270 270 272 274

J - GENERAL SUBSCRIPT
14 14 14 14 20 20 20 20 20
20 20 20 20 20 20 20 36 36
36 36 36 36 110 110 110 120
120 120 122 122 122 122 122
122 124 124 124 124 124 124
124 124 124 124 126 126 126
126 128 128 128 128 128 128
130 130 130 204 204 204 280
280

K - GENERAL SUBSCRIPT
38 38 38 38 38 38 38 58 58
58 58 58 122 122 122 122 204
204 204 204 204

KD - INDEX FOR VARIABLE RANGE IN PLOTTING
76 76 76

KW$ - KEYWORD
198 200 228 228 228 236 238
238

L - LINE LENGTH
56 64 64 66 66

L$ - GENERAL LABEL STRING
8 8 8 8 8 8 8 10 10 10 10
10 10 10 12 16 86 88 130 134
136 136 138 140 142 168 176
186 188

L$(*) - LABEL
16 16 16 186 186 186 262

LA - INDEX FOR LABELING (1=ELEMENTS; 2=NODES)
12 162 164 164 164 166 166
166 166 168 170 174 176 176
176 178 178 178 178

LA$(*) - DESCRIPTOR FOR ELEMENT LABELS
12 168 176 262

LF - SCALE FACTOR FOR PLOTTED LINE LENGTH
70 72 72

LI - LINE INDEX
38 38 38 40 40 40 106 106
106

LL - LINE LENGTH
36 36 68

LN - LABELING OPTION (1=ALL; 2=USER SELECTION)
166 166 166 166 166

LN% - LINE NODE NUMBER ARRAY
214

LN%(*,*) - LINE NODE NUMBER ARRAY
106 272

LO - FLAG FOR OFF SCREEN (1=OFF; 2=NOT)
14 14 26 26 28 28 30 32 38
40 42 76 76 76

NR - # ROWS IN A MATRIX
70 94 94 96 96 98 98 304 304
304 304 304

NS - # SEPARATE BOUNDARIES
108 272 272 274

NU - INDEX
56 56 56 56 56 58 62 64 64

NX - X COMPONENT OF NORMAL VECTOR
60 60 60 64 66 66 66 68 68
68

NY - Y COMPONENT OF NORMAL VECTOR
60 60 60 64 66 66 66 68 68
68

O% - PARAMETER IN &FILL
76 76

O(*,*,*,*) - & ORDER CMD (SORT)
96 210 210 210 304 322 324
326

OP(*) - &FILL OPTION
76 262

P - INDEX IN CONTOUR PLOTTING
16 20 20 20 20 20 20 84 86
88 88 182 182 182 186 312

PB - Y COORDINATE OF PICTURE BOTTOM FOR ZONE PLOT
128 128 130 130 130 132 132
132

PD$ - PROBLEM DESCRIPTOR
236 238

PF - FLAG INDICATING IF CONTOUR POINT FOUND IN AN ELEMENT
14 14 84

PF$ - PICTURE FILE NAMES
308 312

PF$(*) - PICTURE FILE NAMES
308 312 312 314 316 316

PL - X COORDINATE OF PICTURE LEFT FOR ZONE PLOT
128 128 130 130 130 132 832
132

PN - DUMMY PARAMETER
268 268 268 268

PR - X COORDINATE OF PICTURE RIGHT FOR ZONE PLOT
130 130 132 132 132

PT - Y COORDINATE OF PICTURE TOP FOR ZONE PLOT
130 130 132 132 132

PV$(*) - PLOT VARIABLE DESCRIPTOR FOR LABELS (ELEMENT OR NODE)
164 164 164 170 176 176 262
278 278

PX - DUMMY VARIABLE
274 274

PY - DUMMY VARIABLE
274 274

R(*,*) - & RECALL ARRAY
92 96 96 100 208 210 214 216

RD - RADIANS/DEGREE CONVERSION
72 266

RE - REGION # OR RECORD #
90 90 94 98 206 206 208 208
212 212 214 214 222 222 272
272 272 272 292 294 304 306

RS - RIGHT COORDINATE OF SCREEN
146 274

S$ - STARTING
1 112 122 172 190

S0 - SCALE FACTOR FOR FITTING WHOLE PLOT ON SCREEN
146 152 274

S1 - VALUE AT A NODE
20 20 20 84 84 84

S2 - VALUE AT A NODE
20 20 84 84 84

SB - SCREEN BOTTOM Y COORDINATE
26 28 30 82 82 82 88 88 112
128 128 128 130 130 130 154
154 194 194 202 270 274 274
274 322 326 328

SC - SCALE
146 152 154 154 154 154 202
202 202 202 202 274 274 274
274 274 328 328 328 330

SF% - SECTORS FREE
312 312

SI%(*) - STATUS INDICATOR FOR RECORD(0 =UNDEFINED)
90 94 98 206 208 212 214 222 262 292 294

SK - FLAG TO SKIP A SECTION
56 58 58 62 64 64

SL - SCREEN LEFT X COORDINATE
26 28 30 82 82 82 88 88 112
128 128 128 130 130 130 154
154 194 194 202 270 274 274
274 322 326 328

SP - SLOPE OF A LINE
28 28 30 30

SR - SCREEN RIGHT X COORDINATE
26 28 30 82 82 88 88 116 128
128 130 130 146 154 154 194
194 202 270 274 274 322 326

SR$ - 'SELECT A NUMBER' PROMPT
1 120 166 180 196 284 286 294
298 302 314 318

ST - SCREEN TOP Y COORDINATE
26 28 30 82 82 88 88 114 116
128 128 130 130 146 154 154
194 194 202 270 274 274 322
326

SX - X-COMPONENT OF LINE SEGMENT
36 36 36

SY - Y-COMPONENT OF LINE SEGMENT
36 36 36

T - &D(T) SETS TEXT DISPLAY
54 254 282 292 304 306 320

TE - TYPE OF ELEMENT (1=TWO-DIMENSIONAL; 2=AXISYMMETRIC)
236 268

TG - TEMPERATURE GRADIENT MATRIX
98 100 284 294 304

TG(*,*) - TEMPERATURE GRADIENTS (BY COMPONENTS OR MAGNITUDE AND ANGLE)
98 304 304 304

TP - INDICATOR OF PLOT TYPE
102 160 160 160 162 180 282
296 296 296 298 298 298 298
298 298 298 300 300 302 302
302 302 306

TP$(*) - PLOT TYPE DESCRIPTORS (SHADING, CONTOURS, LINES, ETC.)
262 278 278 278 298 300 302
302

TS - TOP OF SCREEN Y COORDINATE
146 274

TV - TEMPORARY VALUE
28 28 28 30 30 30

U$ - UNDERLINE STRING
1 262

UH - HIGH VALUE OF X COORDINATE IN SPECIFIC PLOT
114 116 116 146 154 202 274

UL - LOW VALUE OF X COORDINATE IN SPECIFIC PLOT
116 146 154 202 274 274 274
328

UX - DUMMY VARIABLE
274 274

V - PARAMETER IN &D(V) TO VIEW FULL GRAPHICS
44 112 118 134 166 174 176
182 184 186 190 194 330

V% - VERTICAL INDEX
154 154 154 154

V0 - MINIMUM VALUE
16 70 70 70 76 76 76 80 80
84 96 102 102 102 104 104 104
104 104 104 104 184 186 304

V1 - MAXIMUM VALUE
70 70 70 76 76 96 102 102
102 104 104 104 104 104 104
104 304

VA - GENERAL VARIABLE
94 96 96 96 284 294 304 304
306

VA(*) - VARIABLE BEING PLOTTED
72 72 76 84 84 94 96 96 304
304 304 304 304

VB - BOUNDARY VALUE
56 58 66 66

VH - HIGH VALUE OF Y COORDINATE IN SPECIFIC PLOT
116 116 146 154 202 274

VL - LOW VALUE OF Y COORDINATE IN SPECIFIC PLOT
116 146 154 202 274 274 274
328

VV - VARIABLE VALUE
76 76 76 76

VY - DUMMY PARAMETER
274 274

W(*,*,*,*,*,*) - & WHICH CMD (SELECTS ROWS)
86 322 322 326 326

WR$ - 'WHEN READY' PROMPT
1 4 174 182

X - GENERAL X COORDINATE; X COORDINATE MATRIX
8 8 8 8 10 10 10 10 208 210
210 210 210 328

X(*,*) - X COORDINATES OF NODES
56 56 56 56 204 208 210 210
210 210 210 218 218

X1 - GENERAL X COORDINATE; START OF LINE
36 36 36 36 56 60 60 60 66
68 68 112 116 132 268

X2 - GENERAL X COORDINATE; END OF LINE
36 36 56 60 60 60 66 68 68
68 68 68 68 68 68 68 114 116
132 268

XA - X COORDINATE OFFSET FOR LABEL ON PLOTS
8 8 8 10 10 86 130 130 134
138 140 142 166 174 182 186
190 328 328 330

XC - GENERAL X COORDINATE; CENTROID COORDINATE
38 38 38 40 40 58 58 64 64
70 70 72 72 72 76 76 112 114
218 218 218 218 270 270

XH - MAX X COORDINATE
146 210 274

XL - MIN X COORDINATE
146 154 154 166 178 210 274

XL(*) - X COORDINATE OF LABEL POSITION
12 12 12 48 166 168 176 322
326

XP - GENERAL SCALED X COORDINATE
8 8 8 8 8 10 10 10 10 12
16 44 44 44 44 44 44 46 48
48 48 48 48 50 88 88 88 88
88 88 112 112 112 114 114 116
130 130 134 138 138 140 142
168 176 186 190 190

XP(*) - X COORDINATE FOR NODES OF AN ELEMENT
272

Y - GENERAL Y COORDINATE; X COORDINATE MATRIX
8 8 8 10 10 10

Y1 - GENERAL Y COORDINATE; START OF LINE
36 36 36 36 56 60 60 60 66
68 68 112 116 132 268

Y2 - GENERAL Y COORDINATE; END OF LINE
36 36 56 60 60 60 66 68 68
68 68 68 68 68 68 68 114 116
132 268

YA - Y COORDINATE OFFSET FOR LABEL ON PLOTS
8 8 10 86 130 130 134 138
138 140 142 166 174 182 186
190 328 328 330

YC - GENERAL Y COORDINATE; CENTROID COORDINATES
38 38 38 40 40 58 58 64 64
70 70 72 72 72 76 76 112 114
218 218 218 218 270 270

YH - MAX Y COORDINATE
146 210 274

YL - MINIMUM Y COORDINATE
146 154 154 166 178 210 274

YL(*) -Y COORDINATE OF LABEL POSITION
12 12 12 48 166 168 176 322
326

YP - GENERAL SCALED Y COORDINATE
8 8 10 12 16 44 44 44 44 44
44 44 46 48 48 48 48 48 50
88 88 88 88 88 88 112 112
112 114 114 116 130 130 134
138 138 140 142 168 176 186
190 190

YP(*) - Y COORDINATE OF NODES OF AN ELEMENT
272

Z - USER INTERACTION INDICATOR (0=MAX; 1=INTERMEDIATE; 2=FREE RUN; 3=DEMO)
228 228 254

Z% - FLAGS FOR ZONES SELECTED FOR PLOTTING
120

Z%(*,*) - FLAGS FOR ZONES SELECTED FOR PLOTTING
110 120 120 124 124 124 124
126 126 152

ZI - INDEX FOR ZONES
152 152 154 156 156 156

ZJ - INDEX FOR ZONES
152 152 154 156 156 156

ZN - ZONE NUMBER
42 130 130 130 130 136 136
146 154 154 154 194

ZS - OPTION FOR SELECTING ZONES TO BE PLOTTED
120 120 120 120 122 126

ZS$(*) - DESCRIPTORS FOR ZS
122 262 268 268

ZT - TYPE OF ZONE TO BE PLOTTED
(1=SINGLE;2=MULTIPLE EQUAL-SIZED)
14 42 42 42 42 76 78 136 146
146 150 150 150 152 152 152
152 152 152 152 152 158 194

1 FEB 85 Version JRC/DCD

END OF VAR. LIST

```
*******************************
*                             *
*     DISKETTE PREP.HEAT      *
*                             *
*    TABLE OF VARIABLES       *
*                             *
*******************************
```

A - GENERAL VARIABLE;
32 32 32 32

A$ - GENERAL STRING VARIABLE;
FREQUENTLY INPUT STRING
6 6 6 6 6 6 6 6 30 30
30 30 30 34 46

B(*,*) - & BEEP COMMAND
30 46

B(*,*,*) - & BEEP (WITH REPETITION)
30

BC% - BRANCH CODE (1 ='<' BACKWARD; 2
='>' FORWARD BRANCH; 3 =';' A
PREVIOUS VALUE
8 8 8 8

C(*,*,*,*) - & CLEAR CMD
8

D$ - CHR$(4) CONTROL D
1 6 8 26 26 26 26 28 28 30
36

D(*) - & DISPLAY
(&D(T)=TEXT,&D(G)=MIXED,&D(V)=FULL
GRAPHICS)
1

D1$ - ROW DESCRIPTOR FOR SAVED DATA
8

D1$(*)
8 26 28

D2$ - COLUMN DESCRIPTOR FOR SAVED
DATA
8

D2$(*)
8 26 28

DE$ - DESCRIPTION OF DATA IN
FILEINFO.TXT
8

DE$(*)
8 26 28

DR - DISKETTE DRIVE NUMBER
4 4 6 8 26 30 34

DR$(*) - DISKETTE DRIVE STRING
4 4 6 8 26 30

DS$
8 26

FI$
1 2 26 26 26 26 26 28 28 30

I - GENERAL INDEX
26 26 26 26 26 26 38 38

I(*,*)
4 30 34

I(*,*,*) - & INPUT CMD
46

I(*,*,*,*) - & INPUT CMD
8

I(*,*,*,*,*) - & INPUT CMD
6 30 32

I(*,*,*,*,*,*) - & INPUT CMD
6 8

KW$ - KEYWORD
8 26 28

N - SUBSCRIPT; COMMONLY IN STEPPING
FUNCTION
26 26 28 28

NA$ - FILENAME
8

NA$(*)
8 26 28

RE - REGION # OR RECORD #
28 28 28 28 28 28 28

S - SCALE SIDE #
26 28

T - &D(T) SETS TEXT DISPLAY
1

WR$ - 'WHEN READY' PROMPT
1 30 34

1 FEB 85 Version JRC/DCD

END OF VAR. LIST

```
*******************************
*                             *
*          TEXT.HEAT          *
*                             *
*     TABLE OF VARIABLES      *
*                             *
*******************************
```

A$ - GENERAL STRING VARIABLE; FREQUENTLY INPUT STRING
508

B(*,*) - & BEEP COMMAND
6 508

BC - EQUIVALENT BOUNDARY CONDITION MATRIX
18 18

BC% - BRANCH CODE (1 ='<' BACKWARD; 2 ='>' FORWARD BRANCH; 3 =':' A PREVIOUS VALUE
502 502

BC(*,*) - EQUIVALENT BOUNDARY CONDITIONS
18 18 18

BN% - BOUNDARY NODES
12 12

BN%(*,*) - BOUNDARY NODES ARRAY WITH NODES IN ORDER AROUND BOUNDARIES
12

C(*) - CLEAR ARRAY (&C(*))
6 8 10 10 12 12 14 18

D$ - CHR$(4) CONTROL D
8 8 8 500 500 500

D(*) - & DISPLAY (&D(T)=TEXT,&D(G)=MIXED,&D(V)=FULL GRAPHICS)
2

D1$ - ROW DESCRIPTOR FOR SAVED DATA
500 506

D2$ - COLUMN DESCRIPTOR FOR SAVED DATA
500 506

DE - # NODES / ELEMENT
10 10

DE$ - DESCRIPTION OF DATA IN FNAMES
500 502 506

DR - DISKETTE DRIVE NUMBER
6 8 500 500 504

DR$ - DISKETTE DRIVE STRING
6

DR$(*) - DISKETTE DRIVE STRING
6 6 6 8 500 500 504

F - COUNTER OF # BOUNDARY CONDITIONS ON THE FACE OF ELEMENTS (AS SEEN ON SCREEN)
18

FI$ - 'FILEINFO.TXT'
6 8 8 8 500 500 500

H - NUMBER OF SURFACE CONVECTION BOUNDARY CONDITIONS
18

I - GENERAL INDEX
2 2

I(*,*) - &INPUT CMD
508

I(*,*,*,*,*,*) - & INPUT CMD
502

KW$ - KEYWORD
8 8 502 502 502

LN% - LINE NODE NUMBER ARRAY
12 12

LN%(*,*) - LINE NODE NUMBER ARRAY
12

MP - MATERIAL PROPERTY MATRIX
14 14

MP(*,*) - MATERIAL PROPERTY MATRIX
14

N$ - GENERAL STRING
10 10 12 12 14 18 500 504

N1 - GENERAL INDEX; NUMBER OF ROWS IN SAVED ARRAY
10 10 12 12 18 500 506

N2 - GENERAL INDEX; NUMBER OF COLUMNS IN SAVED ARRAY
10 14 500 506

NA$ - FILENAME
500 500 502 502 502 502 502
504 510

NB - # OF BOUNDARY NODES IN BN%
12 12

ND% - ELEMENT NODE #S
10 10

ND%(*,*) - ELEMENT NODE #S
10

NE - NUMBER OF ELEMENTS
10 10 14

NL - # UNIQUE LINES
12 12

NN - NUMBER OF NODES
10 10

NP - NUMBER OF DATA POINTS
18 18

PD$ - PROBLEM DESCRIPTOR
8 8

R(*,*) - & RECALL ARRAY
10 10 12 12 14 18

RE - REGION # OR RECORD #
10 10 12 12 14 18 500 500
500

SI% - STATUS INDICATOR
8

SI%(*) - STATUS INDICATOR FOR RECORD(0 =UNDEFINED)
8 500 500

T - &D(T) SETS TEXT DISPLAY
2

TE - TYPE OF ELEMENT (1=TWO-DIMENSIONAL; 2=AXISYMMETRIC)
8

XY - COORDINATES OF POINTS
10 10

XY(*,*) - COORDINATES OF POINTS
10

1 FEB 85 Version JRC/DCD

END OF VAR. LIST

2.4.2 Line Cross-References

```
******************************
*                            *
*        HELLO.HEAT          *
*                            *
*   TABLE  OF  LN#  XREFS    *
*                            *
******************************

    8
4

    10
8  20

    12
14

    14
12  12  16

    36
38

    38
34

1  FEB  85  Version  JRC/DCD

END  OF  LINE  LIST

******************************
*                            *
*       GEOMETRY.HEAT        *
*                            *
*   TABLE  OF  LN#  XREFS    *
*                            *
******************************

    4
6  64  146  148  176  190  200  220
248  258  262  272  288  302  306  308
312  314  316  318  320

    6
106  116  132  138  270

    8
108

    10
10  10

    12
64  76  90  94

    14
52  54

    16
54

    18
16  56

    20
10  102  102

    22
20

    24
122

    26
122

    28
40  126  220

    32
126  220

    34
10  36  64  76  76  90  90  94  94  94
98  100  102  102  102

    36
124  126

    38
118  124

    40
130

    42
144

    44
48

    46
44  48

    50
84  86  88  102

    52
44  46

    54
52  52

    58
12  26  28  28  32  40  50  54  106
106  108  116  178  218  228  238  242
248  250  260  294  296  296  298  298
300  300  302  302  304  304  304  316
316  318

    60
62  64  110

    66
66  76  84  90
```

68
98 100

70
98 100

72
92

74
74

76
72

78
96

80
80 82 82 86 110

84
80

86
80 86 88 110

90
86

92
72 72 72 74 74 76 92 94 96

96
76 76 96 96 96

100
96

102
96 96 96 98

104
106 300

108
10 60 62 62 64 80 80 82 86 88
102 108 110 110 110 154

110
108 110

112
154 154

114
106 116 118 140 296

118
118 130

120
122 122 122 124

124
122

126
124

128
116 118 118 130 130 132

132
116 128 162 162

134
140 148 148

142
144 146

146
144

148
140 146 172 172

150
172 294

152
152 152 152 152 154

154
108 108 152 154 154 294

156
172 296

158
160 162

162
132 162 162

164
132 172

166
166 166 166 166 166 168

168
166 166 166 168 168

170
148 148 154 162 168 168 172 172
172 296

174
132 178

178
174 176

180
112

182
184

184
182

186
148

188
190

190
188

192
114 294 296 302 316

194
194

196
296 304 316

198
198

200
192 196

202
204 204 208 210 210 214 214 214

204
202

206
106 116 138 318

218
106 116 138

220
40 40 116 318

222
296 300 318

224
180 186 192 196 254

226
182 188 226

228
184 190

230
180 192 196 274

232
232

234
194 198 234

236
226 234

238
226 234

240
226 234

242
194 198

244
246 274

246
302 316

248
274

250
184 190

252
180 186 192 262

254
252 258 258 258

256
252 286

258
252

260
258

262
230

264
1 274

274
274 274 274 284

282
2 284 284 286

286
284 284 288

288
282

292
284 284

```
    294
284 284 298

    298
298 298 298 300

    300
300 300 300

    302
284

    306
302

    308
304

    310
178 304

    314
304

    316
284

    320
274

1 FEB 85 Version JRC/DCD

END OF LINE LIST


          ******************************
          *                            *
          *         GRID.HEAT          *
          *                            *
          *    TABLE OF LN# XREFS      *
          *                            *
          ******************************


     4
36 38 48 98 118 132 154 156
160 162 162 166 166 170 174 178
186 188 190 194 196 234 238 244
248 254 264 288 294 306 314 324
332 334

     6
114 188

     8
10

    10
332

    12
10

    14
16

    18
14 18

    20
4 8 18 18 18 50 50 72 98 120
126 138 138 140 150 150 190 194
196 206 208 222 222 224 236 270
272 276 278 278 288 290 294 296
308 312 314 318 320 322 324 332
334 336 336 338 338 342 348 350
352

    22
168

    24
12 22

    28
24

    30
172 178

    40
298

    50
46

    54
44

    56
56 192

    58
58 196

    60
62 334

    64
62

    70
66

    72
66

    74
286

    86
84

    88
84

    90
84
```

222
138 140 150 152 312 314

224
124 130 302

226
120 126 136 148 182 184 298 310

228
226 234 234 234

230
226 262

232
156 226

234
226

236
234

238
210

240
1

256
2 260 262

260
258

262
258 260 262 264

264
256

268
4 118 258 272

270
260

272
270 272

274
260

278
272 276 278 278 280 282

280
278 280 280

282
280 282 282

284
280 282

288
260 272 290 290

292
290

294
260 272 296 296

298
296

300
302

302
300

304
306

306
304

308
260

310
260 314 314

312
312

314
314

316
310

318
260

320
320 346

322
324 346

330
332 352

334
352

336
276 338

338
336 338 338 338

```
    342
348

    346
346

    348
346

    350
276

1 FEB 85 Version JRC/DCD

END OF LINE LIST
```

```
            ******************************
            *                            *
            *      PREPROCESS.HEAT       *
            *                            *
            *     TABLE OF LN# XREFS     *
            *                            *
            ******************************
```

```
    4
6 86 106 174 208 210 220 252
272 290 312 318 322 330 332

    6
92 96 112 142 216 260 280 282

    8
8 18 20 20

    10
18 20 22 22 24 28

    12
10

    14
18 20 22 22 24 28

    16
10 18 18 20 20 22 22 24 24 26
28 28 32 38 40 94 112 112 138
162 162 164 170 172 182 192 196
200 256 256 268 276 276 294

    18
130 132

    20
130 132

    26
26 28

    28
130

    30
146

    34
144 146

    36
30

    40
208

    42
12 14 60

    44
334

    46
42 330

    48
42

    50
80 112 306 322 330

    54
80 86 100

    56
44

    60
112 322

    62
54 64 86 86 106 106

    64
100

    66
72 94 94 262

    68
94

    70
72 72 72 76

    74
72 90

    76
76 90

    80
78

    82
80 82 82 82 84 84

    84
82
```

86
82 82 84

88
74 88 88 92 106

92
72 72 76 78 88 88

94
96 262

96
94

98
80 100

100
80

102
100 102 102 102 104 104

104
102

106
102 102 104

108
136 282

110
108 112 114

114
114 116 116 118 120 120 122 122
124 124 126 126 128 130 130 132
132 132 134

116
116 120 122 124 126 128

122
118 122

124
118 118 118 120 124 132

126
126 126 126

128
128

130
126 128

132
132

134
114 114 134

138
136

140
142

142
140

148
150

150
148

152
232

160
162 164 164 170

162
162

164
164

168
170 172 172

170
170

172
172

174
160 162 168 170 258 278 326

176
174 202

178
138 146 160 162 168 170 204 258
262 272 278 326

180
140 148 180 264

182
142 150 266

184
138 146 160 222 258 262 272 326

186
186

188
162 164 170 172 188 260 280 328

190
180 188

192
180 188

194
180 188

196
162 164 170 172 260 280 328

198
222

200
150 266

202
138 146 160 258 262 278 326

204
202 202 208 208 208

206
202 210 248

208
202

210
184

212
1 244

222
218

238
2 244 244 248

246
238 244 244 252

256
244 256 272 276

258
284 292

260
260

262
256

264
266

268
264

270
244

278
308 320

280
280

282
276

284
244

286
260

292
244

294
260

298
296

302
302

304
302

306
298

308
244

310
280

314
244

316
330

320
244

322
280

324
244

326
314

328
328

332
316 318 330

334
330

```
   336
330

1 FEB 85 Version JRC/DCD

END OF LINE LIST


          *****************************
          *                           *
          *        SOLVE.HEAT         *
          *                           *
          *    TABLE OF LN# XREFS     *
          *                           *
          *****************************

      4
6 52 62 64 68 84 86 94 112
116 122 126 132 136 138 150 188
190 198 198 208 222

      6
26 38 46 50 140 142

      8
32 40

     10
42

     12
124

     16
4 6 10 26 32 124 134 156 158
158 168 172 172 178 192 230 234

     18
102

     24
18

     26
42 44

     28
44

     32
14 48 126

     34
126 126

     36
30

     40
64 218 232

     48
44

     56
58

     58
56

     60
62

     62
60

     64
86 218 232

     66
64

     68
68

     70
64

     72
70

     76
72

     78
72 76

     82
84

     84
82

     86
92 116 218 234

     88
86

     90
90

     92
86

     94
94

     96
92

    100
72 72 78 78 98 98 102

    102
96 98

    106
108
```

108
106

110
112

112
110

114
218 234

116
120

118
114

120
120

122
122

124
114

130
132

132
130

134
218

136
136

138
134

140
50 52 100

142
50 52 100 102

144
202

146
146

148
148

150
134 144 146

152
54 58 64 80 86 92 104 108 114
120 128 134 144 146 182 202 202
204 206

154
56 60 82 106 110 130 154

156
58 62 84 108 112 132

160
54 64 80 86 92 104 114 128 134
144 200

162
162

164
68 90 94 120 122 136 146 148
164 206 208

166
152 154 154 154 164 164 170

168
154 164

170
154 164

172
68 90 94 120 122 136 146 148
206 208

174
176

176
200

178
62 84 112 132

180
54 66 80 88 92 104 118 128 134
144 202

182
180 180 188 188 188

184
180 190 202 220

186
180 190

188
180

190
160

192
188

194
1

```
     200
198

     206
206

     208
208

     210
2  218  220

     218
216

     220
218  222

     222
210

     228
4  232  232  234  234

     230
218

     232
230

1 FEB 85 Version JRC/DCD

END OF LINE LIST


        *****************************
        *                           *
        *     POSTPROCESS.HEAT      *
        *                           *
        *   TABLE OF LN# XREFS      *
        *                           *
        *****************************


      1
182

      2
148

      4
20  22  28  32  36  42  50  64  68  76
76  86  92  96  100  110  150  152
160  162  186  200

      6
60

     10
4  20  32  50  58  58  70  122  124
124  138  138  154

     12
22

     14
34  52

     16
34

     18
182  198

     20
20  20  20

     22
20

     26
28

     28
26
     30
182  198

     32
32  32  32

     34
32

     42
40

     44
32  42  58  76

     46
36  64  76

     48
50  50  182  198

     50
50

     52
50

     56
52

     60
58

     66
68

     68
66

     70
76  182

     76
76
```

78
76

80
76

82
182 198

84
84

86
86

92
90

94
96

96
94

98
182 198

100
100 100

108
110

110
108

112
166

114
114

116
116

118
18 24 30 38 48 64 72 82 84 92
98 100 106 112 114 146 166 166
192

120
26 40 66 94 108 120

122
28 42 68 96 110

126
18 24 30 38 48 64 70 82 98
100 112 166 192

128
128

130
20 32 50 76 84 86 100 100 114
116 130 194

132
120 120 130 136

134
120 130 132

136
120 130

138
20 32 50 76 84 86 100 100 114
116 194

140
142

142
166

144
18 24 30 38 48 64 72 82 92 98
112 192

146
144 144 150 150 150

148
144 152 184

150
144

152
126

154
150

156
144 152 200

158
1

164
162

176
2 182 182 184 196 198 198 198

184
182 182 184 186

186
176

190
4 58 198

192
170

194
194

```
     196
182

     198
182  198

     200
18  30  48  72  82  84  98  100  112
114

1 FEB 85 Version JRC/DCD

END OF LINE LIST

                ******************************
                *                            *
                *         PLOT.HEAT          *
                *                            *
                *     TABLE OF LN# XREFS     *
                *                            *
                ******************************

     1
284

     4
92  96  100  112  144  148  170  174
194  198  200  220  246  248  256  260
288  308  310

     6
44  48  134

     8
176  186  192

     10
12  18  48  48  88  88  114  130  130
134  134  138  138  140  142  168  190

     12
12  12  174

     14
20  22

     16
16  18  182

     20
86

     22
86

     24
14  14  20  26  40  50  68  86  86
102  102  104  104  112  112  118  118
120  126  136  144  150  150  152  152
152  158  180  196  246  250

     26
14  38  42

     32
28

     34
14  42  42

     36
42  42

     38
76

     42
62  72  106  108

     44
46  112  114  190

     48
12  18  48  114  190

     52
86  112  118  120  122  150  154  158
162  164  176  180  186  188  190  192
194

     54
144  146  164  172  196

     56
162

     58
56

     60
56

     62
58  64  64  66

     64
56

     66
60

     68
72

     70
160

     76
160

     82
160  160

     84
160

     90
292
```

92
92

94
306

96
96

98
304

100
100

102
96 104 304

104
104 104 104

106
160

108
160

110
124

112
114 116 152

118
118 118 120 152

120
120

124
126

128
118 118 120 120

132
116 116

134
116 118 134

136
78 160 162

138
136

140
80 80 184 184

142
76 78 84 86

144
158 292 304 306

150
150 152

154
154

156
152 154

158
146 152

160
146 152 154

164
178 180 188

170
166

172
166 176

176
168

178
164 170 176

180
166 180 182 182

188
180 180 182 186 188 188 188 192
192

192
190

194
162 188 188 188

196
196 198 198 200

198
198

202
116 146 274

204
274

206
272

208
208

210
210

214
214

216
216

220
90 94 98 206 208 212 214

222
90 94 98 206 208 212 214 242
272 272 272 272

224
90 94 98 206 262

226
226

228
92 96 100 208 210 214 216 228

230
222 228 228 232

232
228

234
92 96 100 208 210 214 216

236
238

238
262

240
90 94 98 206

242
240 246 246

244
240 248 272 286

246
240

248
224

250
246

252
220 240 248

254
1

262
258

282
2 284 286 292 292 294 304 306
306 308 310 314 314 316 318 318

286
284 284 288

292
284

294
284 298 304

300
296

302
300 306

304
300

306
284 306

308
284 314 318 318

310
308

312
310 318 320

314
312

316
316

318
242

322
166

326
166

328
160

332
258

1 FEB 85 Version JRC/DCD

END OF LINE LIST

```
          *******************************
          *                             *
          *      DISKETTE PREP.HEAT     *
          *                             *
          *      TABLE OF LN# XREFS     *
          *                             *
          *******************************


    2
8 32

    10
8

    32
6

    34
32

    36
34

    38
2 2

    40
4

    42
40

    44
40

    46
42 44

    48
40

1 FEB 85 Version JRC/DCD

END OF LINE LIST


          *******************************
          *                             *
          *          TEXT.HEAT          *
          *                             *
          *      TABLE OF LN# XREFS     *
          *                             *
          *******************************


    500
10 10 12 12 14 18

    502
502

    504
502

    506
10 10 12 12 14 18

    508
6 6 8

    510
6

*00108
500

1 FEB 85 Version JRC/DCD

END OF LINE LIST
```

3.0 ELASTICITY

This chapter deals with applications of the elasticity program.

Chapter 2 is not a prerequisite for this chapter. However, if you have read the preceding chapter, you will notice many similarities; and you should be able to move quickly through this chapter.

Actually, the GEOMETRY and GRID programs which deal solely with the geometric aspects and discretization are identical in each chapter. Refer to Chapters 22 through 24 of Segerlind's text if you need to review the theory of elasticity. Here we concentrate entirely on the procedural aspects of the finite element elasticity program. As with the heat conduction program, we treat both the two-dimensional and the axisymmetric cases using the three node triangular element.

3.1 Elasticity Example: Axisymmetric Column and Footing

We assume that you have an Apple II (+, -e, or -c) computer with a monochrome monitor and two disk drives. If you have only one drive, you must follow the diskette swapping instructions carefully. You need only a minimal familiarity with the operating characteristics of the Apple II to follow these instructions.

First, make a working copy of the Elasticity Program Diskette, store the original in a safe place, and write-protect the working copy. In addition to the copy of the program diskette, you need a blank data diskette. Since diskettes are fragile, protect them from dust and fingerprints.

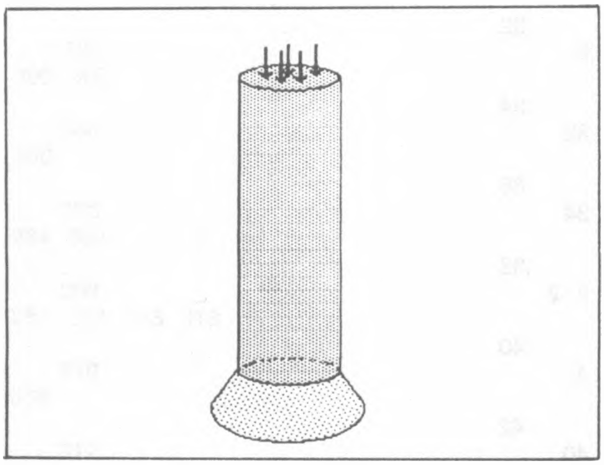
Here is a statement of the problem:

> An axisymmetric wooden column resting on a tapered concrete footing is subjected to a pressure from above (Gustafson 1977, 74). Obtain the stress-strain results.

As an important first step, make a sketch of the problem (fig. 3.1) to guide your analysis. Because of the problem symmetry, you need consider a cross-section of the body of revolution.

To begin the program, place the program diskette in drive 1 and turn on the monitor and computer. After several startup messages, you must specify the number of disk drives you are using. If you have two, press 'return' to accept the default shown in inverse video enclosed in square brackets. If you have only one drive, you must have the program diskette in the drive when you change program segments; but the data diskette must be resident at all other times with one exception in PLOT.

Fig. 3.1 Axisymmetric wooden column on a tapered concrete footing subjected to a vertically downward loading (Gustafson 1977, 74)



Next specify the desired level of user interaction. If you do not fully understand the ramifications of this question, request additional information by entering a question mark as the first character after the default and pressing 'return'. 'HELP 0.4' refers you to the ELASTICITY Help Messages found in section 3.3 of this book. Review the help message now.

The first digit of the help message reminds you of the identity of the current program segment, and the number after the period identifies the current input statement. The printed help message provides background and identifies the source code line number should you wish to refer to section 3.4. If you need additional background on the finite element method, page or chapter references to the companion book by Segerlind are given. For this example, choose the intermediate level of interaction. Press 'return' when you are ready to resume.

The title and copyright notice follow (fig. 3.2).

Fig. 3.2 Elasticity program title and copyright notice

```
APPLIED FINITE ELEMENT ANALYSIS:
AN APPLE ][ IMPLEMENTATION

ELASTICITY

BY

J. R. COOKE AND D. C. DAVIS

COPYRIGHT, 1985

JOHN WILEY AND SONS, INC.

ALL RIGHTS RESERVED
```

Press 'return' again to reach the most important menu (fig. 3.3) in the entire program, referred to throughout as the MAIN MENU.

Fig. 3.3 The MAIN MENU

```
***     MAIN PROGRAM MENU (0.00)     ***

1. GEOMETRY
   DEFINE PROBLEM GEOMETRY
2. GRID
   GENERATE & MODIFY GRID
3. PREPROCESS
   DEFINE PROPERTIES AND BOUNDARY
   CONDITIONS
4. SOLVE
   ASSEMBLE EQUATIONS & SOLVE FOR NODE
   DISPLACEMENTS
5. POSTPROCESS
   SOLVE FOR ELEMENT STRAINS & STRESSES
   & NODE STRESSES
6. PLOT
   PLOT GRID, BOUNDARY CONDITIONS, &
   PROBLEM RESULTS
7. DISKETTE PREPARATION
   PREPARE DATA DISKETTE

0. STOP
```

Table 3.1 outlines the overall program organization and the function of each of the program segments. To solve the problem proceed sequentially through the first six items. The default prompt keeps track of your progress. If you skip one of the items, certain intermediate results needed to complete a subsequent step will not be generated. On the other hand, if you return to an earlier program segment, the program disables all data files dependent upon the revised file you create. This safety feature assures that your data files are internally compatible.

DISKETTE PREPARATION

Since this is your first exercise, you must prepare a data diskette before you can proceed with the problem formulation. We made DISKETTE PREPARATION the last executable item in the menu to reduce the possibility of accidental diskette erasure. Select option 7 and press 'return'. Insert the data diskette as instructed. Be sure that you write-protect the working copy of the program diskette, since a wrong move here could be disappointing. Press 'return' when ready.

This is your last opportunity to review the current contents of the data diskette before you must decide whether to reformat. As a safety step, you must authorize diskette erasure with 'Y' and 'return'. The diskette produced has 543 free sectors, 47 more than you would have gotten with the usual INIT command.

Provide an identifying descriptive keyword of up to 8 characters as indicated by the number of underscores. Choose "ELAST1" to remind you of 'elasticity example 1'. This keyword will be the prefix for each of the related data files. Commas and colons are prohibited here and in the more complete description which you enter next. Press 'return' and wait for the directory 'FILEINFO.TXT' to be created. Table 1.4 describes the file contents. Return to the MAIN MENU.

Problem formulation always begins at the MAIN MENU to assure the proper configuration of memory. Re-select the intermediate level of user interaction. This intentionally repeated question allows you to return to the MAIN MENU at any point and change this option.

Select GEOMETRY and press 'return' and the program segment will be loaded and activated.

GEOMETRY

GEOMETRY begins with an abstract (fig. 3.4).

Fig. 3.4 GEOMETRY abstract

```
APPLIED FINITE ELEMENT ANALYSIS:
AN APPLE ][ IMPLEMENTATION

COPYRIGHT, 1985

GEOMETRY (1.00)

ABSTRACT:

DISCRETIZATION BEGINS WITH THE CREATION
OF CURVILINEAR QUADRILATERAL SUB-REGIONS
WHICH TOGETHER DESCRIBE THE OVERALL
GEOMETRY.

SUBSEQUENTLY, THESE INPUT REGIONS WILL
BE SUB-DIVIDED INTO A MESH OF 3-NODE
TRIANGLES.
```

The program library contains only two element types. Select the axisymmetric 3-node triangle for this problem.

Each program segment has a principal menu numbered (X.05) to guide your progress (see fig. 3.5). Select 'input with graphics'. Since you are not revising or reviewing an existing data file, you cannot retrieve the data. If such a data file did exist and were retrieved, the defaults would reflect this data.

Fig. 3.5 GEOMETRY principal menu

```
***          GEOMETRY (1.05)          ***

 --- POINT COORDINATE ENTRY ---
1. INPUT WITH GRAPHICS
2. INPUT WITHOUT GRAPHICS

 ---  REGION DEFINITION/DIVISION  ---
3. INPUT WITH GRAPHICS
4. INPUT WITHOUT GRAPHICS

 ---  DATA REVIEW   ---
5. LIST INPUT DATA
6. PLOT INPUT DATA

0. NONE OF THE ABOVE
```

Enter the data shown in figure 3.6. As indicated, establish the global coordinate system with 0 <= R <= 0.5m and 0 <= Z <= 2.5m. If you enter a value incorrectly, use the arrow keys to edit the input before you press 'return'. If you do not discover an input error until after pressing 'return', use '<' as the first input character after the default and press 'return' to go back to the previous question. If you wish to abandon your current formulation, enter control-Q as the first character without pressing 'return'; and you will be returned to the principal menu to restart this step.

Enter (E) the 13 points needed to define the two mesh generating regions shown in figure 3.6. Enter the points individually or with straight line segments. You can generate the mid-side points automatically using the Line (L) option. Since a uniform mesh will be adequate, place the mid-side point at the middle of each side of the quadrilateral. Select 'Q' to leave the input of points.

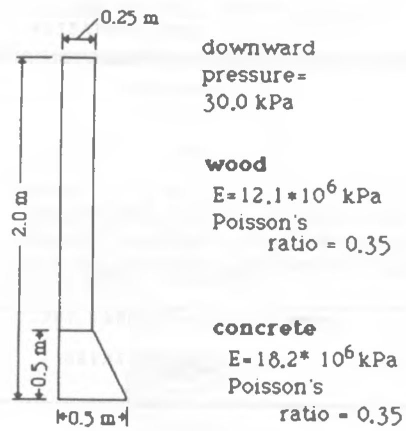
From the GEOMETRY menu proceed to region definition with graphics. Beginning at a vertex, create region 1 by defining a COUNTERCLOCKWISE sequence of 8 points as suggested by Figure 3.7. After you specify all eight points, exit with 'Q' and visually verify that you have followed the counterclockwise rule (see fig. 3.7). Next define region 2 with a counterclockwise sequence. Three of the points are shared with region 1. All regions have now been defined. Use 'Q' to quit this code segment.

The program automatically identifies common region boundaries. To complete the creation of the mesh (fig. 3.8), designate the number of element nodes to be placed along each side of the mesh generating region. Point to a side by moving the 'X' cursor to it and then select it by pressing 'S'. Enter the number of nodes per side as shown in figure 3.8.

When you generate a mesh, remember that the elements in a mesh are connected to other elements only through shared nodes (i.e., vertices of the triangles). Consequently, when you work with multiple mesh generating quadrilaterals, the common boundaries must have the same number of element nodes. The program automatically assures this condition if the automatic mesh generator is used.

Review the data generated in this program segment (options 5 and 6) before selecting option 0 (fig. 3.5) to proceed to GRID generation.

Fig. 3.6 Axisymmetric column problem formulation




```
GEOMETRY: COORDINATES (1.31)

INSTRUCTIONS:

ENTER POINTS TO BE USED TO DEFINE 8-NODE
MESH GENERATING REGIONS WITH 4 VERTICES
& A NODE ON EACH OF THE 4 (STRAIGHT OR
CURVED) SIDES. A TRIANGULAR REGION MAY
BE CREATED IF TWO CONNECTED STRAIGHT
LINES FORM ONE SIDE.   ADJACENT REGIONS
MUST SHARE 3 POINTS ON THE COMMON SIDE.
```

example

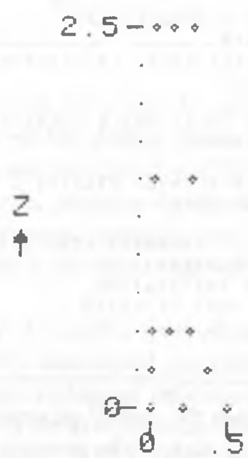
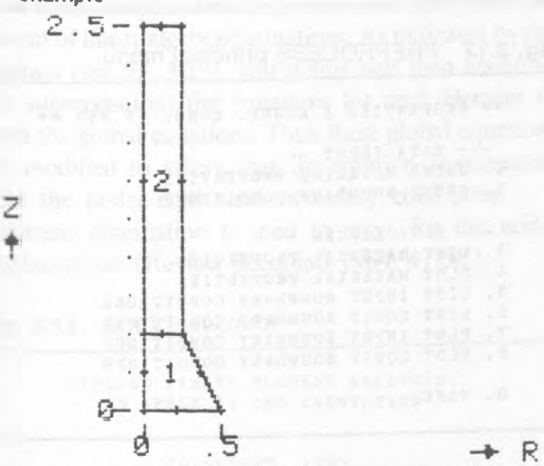


Fig. 3.7 Mesh generating regions

```
INPUT OF POINTS-
 OPTIONS: ENTER, DELETE, QUIT, BRANCH
( E D Q < > )[E]?

GEOMETRY: REGION DEFN'S (1.32)

INSTRUCTIONS:

BEGINNING WITH A VERTEX, DEFINE EACH
REGION WITH A COUNTER-CLOCKWISE SEQUENCE
OF 8 POINTS.

USE ARROWS TO POSITION THE CURSOR. THEN
PRESS <S> TO SELECT.

EACH REGION IS NUMBERED AT ITS CENTROID.
```

example



GRID

The abstract reminds you of the purpose of this program segment (fig. 3.9) . You can execute this program segment in a local free-run mode using option 1 (fig. 3.10), or you can be more deliberate and retain your ability to examine internal details by sequentially performing options 2-4 individually. On this first example, select option 1 and run the program segment without user interaction. The program retrieves data from GEOMETRY and calculates the nodal coordinates.

The mesh for each region is plotted separately and then the composite mesh is drawn. The mesh has 72 elements and 52 nodes. The 'bandwidth' of 50 stipulates that the maximum difference in related node numbers is 49 (i.e., one less than the number given). (See Segerlind 1984, 47-50 for a more detailed discussion.) At this point you need only recognize that this number is related to the width of the largest array you will generate and, hence, provides a clue to the degree of mesh refinement this program can handle.

Fig. 3.8 Number of nodes on each side of the mesh-generating region

```
GEOMETRY: NODES PER SIDE  (1.33)

INSTRUCTIONS:
   THE  NUMBER  OF  TRIANGULAR   ELEMENT
NODES ON EACH  CURVILINEAR QUADRILATERAL
REGION SIDE (INCLUDING BOTH  END  NODES)
IS >=2. OPPOSITE REGION SIDES MUST HAVE
THE SAME NUMBER OF NODES; THIS IS AUTO-
MATICALLY ENFORCED.

USE ARROWS TO POSITION THE CURSOR. THEN
PRESS <S> TO SELECT.

ENTER THE NUMBER OF NODES FOR THE MARKED
SIDES. CHANGE ENTRIES BY RE-SELECTING
THE SIDE.
```

example

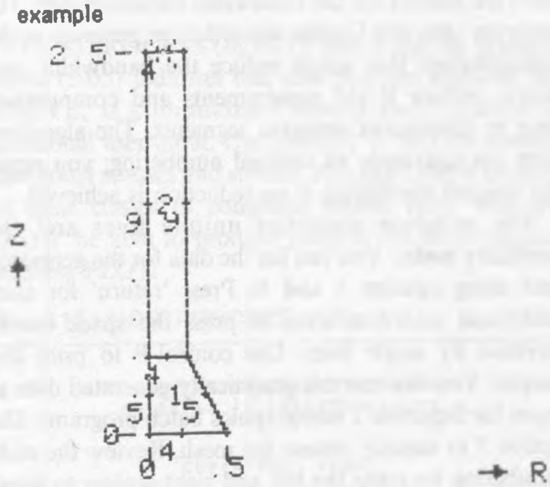


Fig. 3.9 GRID abstract

```
        APPLIED FINITE ELEMENT ANALYSIS:
          AN APPLE ][ IMPLEMENTATION

                COPYRIGHT, 1985

                  GRID  (2.00)

ABSTRACT:

A GRID OF  3-NODE TRIANGULAR ELEMENTS IS
GENERATED FOR THE PROBLEM DOMAIN.  NEXT,
NODES ARE RENUMBERED  TO REDUCE  PROBLEM
BANDWIDTH.  THEN UNIQUE LINES AND  NODES
ON BOUNDARIES ARE IDENTIFIED.
```

The current formulation certainly does not tax memory limitations.

Here, or at any input request, examine memory usage by entering control-A immediately after the default bracket (without pressing 'return'). Note that more than 9K of RAM is unused at this point. Typically, RAM does not limit this step. 'Related nodes' (i.e., those nodes connected to the node in question by a single element

Fig. 3.10 GRID principal menu

```
        ** GRID  (2.05) **

      -- GRID DEFINITION --
    1. DO COMPLETE GRID GENERATION (2-4)

    2. DEFINE GRID ELEMENTS & NODES
    3. RENUMBER NODES TO REDUCE BANDWIDTH
    4. SORT UNIQUE LINES & BOUNDARY NODES

      -- GRID EXAMINATION --
    5. LIST ELEMENT NODAL COORDINATES
    6. LIST LINE & BOUNDARY NODES
    7. PLOT ENTIRE GRID

    0.NONE OF THE ABOVE
```

side) are needed for the bandwidth reduction step. The program uses the Collins algorithm to generate nodal renumberings that might reduce the bandwidth and, hence, reduce RAM requirements and computation time in subsequent program segments. The algorithm does not guarantee an optimal numbering; you retain the original numbering if no reduction is achieved.

The program identifies unique lines and the boundary nodes. You can list the data for the generated grid using options 5 and 6. Press 'return' for each additional screen of data or press the space bar to advance by single lines. Use control-P to print this output. You can use this graphically generated data as input for Segerlind's non-graphics batch programs. Use option 7 to visually review the mesh. Review the node numbering by using the left and right arrows to move the cursor. You need identify only selected nodes to use with the results yet to be calculated. You can relegate the numbering of the interior points to a secondary role because of the graphical support.

Now you have completed the geometrical formulation and can proceed to PREPROCESS to the input of the material properties and the boundary conditions. Select item 0 of the principal menu and proceed to the exit menu and then to PREPROCESS. (If you were dealing with a two-dimensional formulation, the thickness input would also be done in the next program segment, PREPROCESS.)

PREPROCESS

The program (fig. 3.11) retrieves data generated in the previous program segments, generates additional information, and presents the principal menu (3.05). (See fig. 3.12). As before, the menu consists of data input and data review sections. Proceed sequentially through options 1 and 2.

Assume that the elastic modulus (Young's modulus) is constant everywhere within each region so choose to enter the values by region (option 2). You may use coded values to enter the properties for each region.

Fig. 3.11 PREPROCESS abstract

```
          APPLIED FINITE ELEMENT ANALYSIS:
             AN APPLE ][ IMPLEMENTATION


                  COPYRIGHT, 1985

                 PREPROCESS (3.00)

ABSTRACT:
SPECIFY 'MATERIAL' PROPERTIES AND BOUND-
ARY CONDITIONS, IN THAT ORDER.
THE REQUIRED PROPERTIES ARE THE ELASTIC
MODULUS, POISSON'S RATIO, THICKNESS,
TEMPERATURE AND THERMAL EXPANSION
COEFFICIENT FOR EACH  ELEMENT.

BOUNDARYCONDITIONS INCLUDE POINT FORCE,
FIXED   NODAL DISPLACEMENT, AND SURFACE
STRESS.
```

Fig. 3.12 PREPROCESS principal menu

```
    ** PROPERTIES & BOUND. COND. (3.05) **

     --- DATA INPUT --
    1. ENTER MATERIAL PROPERTIES
    2. ENTER BOUNDARY CONDITIONS

     --- DATA REVIEW --
    3. LIST MATERIAL PROPERTIES
    4. PLOT MATERIAL PROPERTIES
    5. LIST INPUT BOUNDARY CONDITIONS
    6. LIST EQUIV BOUNDARY CONDITIONS
    7. PLOT INPUT BOUNDARY CONDITIONS
    8. PLOT EQUIV BOUNDARY CONDITIONS

    0. NONE
```

Assign A = 18.2×10^6 kPa and B = 12.1×10^6 kPa. After each region is identified for you, select A or B. You can repeat this process for Poisson's ratio (0.35); or since you assume both materials have the same value, you could enter a single value for the entire body. If you were solving a two-dimensional problem, the program would request the thickness data. Next the program requests the temperature and the coefficient of thermal expansion. Since you are considering an isothermal problem, simply enter zero for these two properties. The reference temperature for thermal expansion calculations is also zero.

You have one more opportunity to edit the properties before proceeding to input the boundary conditions. This opportunity is especially useful if a property is nearly uniform; if so, enter the uniform property and edit the exceptions.

Although you can specify the boundaries at individual nodes or for a continuous group of nodes, use the latter. Enter a zero R-displacement along the axis of symmetry and a zero Z-displacement along the base of the concrete. Enter the surface stress along the top as a negative Z or negative normal value. No condition need be imposed on the outer (rightmost) surface, because a traction free condition is automatically assumed. Use 'Q' to quit. The program computes the nodal equivalent of the boundary conditions before returning you to the principal menu.

Review the properties and boundary conditions using option 3-8. You can list and plot the material properties. Shading is used to plot the assignments for properties. Input boundary conditions refer to your input. Equivalent boundary conditions refer to the nodal equivalents of the conditions you imposed; this form of the boundary conditions will be used in the SOLVE program. Proceed to SOLVE.

SOLVE

You have completed the input and can now form the system of linear algebraic equations. As indicated by the abstract (see fig. 3.13), you define and then assemble (by superposition) the equations for each element to form the global equations. Then these global equations are modified to assure that the solution is consistent with the nodal equivalent boundary conditions. Gaussian elimination is used to solve for the nodal displacements (Review Segerlind 1984, 47-50.)

Fig. 3.13 SOLVE abstract

```
        APPLIED FINITE ELEMENT ANALYSIS:
           AN APPLE ][ IMPLEMENTATION


               COPYRIGHT, 1985

                 SOLVE (4.00)

ABSTRACT:

FINITE ELEMENT EQUATIONS ARE DEFINED FOR
EACH ELEMENT, ASSEMBLED INTO A GLOBAL
MATRIX EQUATION, MODIFIED BY BOUNDARY
CONDITIONS, AND SOLVED FOR NODAL
DISPLACEMENTS.
```

Fig. 3.14 SOLVE principal menu

```
*** PROBLEM SOLUTION (4.05) ***

 -- FORM & SOLVE ALL EQUATIONS --
1. DO COMPLETE PROBLEM SOLUTION
   (MENU OPTIONS 2 - 5)

 -- FORM SYSTEM EQUATIONS --
2. ASSEMBLE GLOBAL FORCE AND STIFFNESS
   MATRICES
3. APPLY EXTERNAL FORCE BOUNDARY
   CONDITIONS
4. APPLY DISPLACEMENT BOUNDARY
   CONDITIONS

 -- SOLVE SYSTEM EQUATIONS --
5. SOLVE FOR NODE DISPLACEMENTS

 -- LIST RESULTS --
6. LIST NODE DISPLACEMENTS

0. NONE OF THE ABOVE
```

The principal menu (4.05) allows you to execute options 2-5 without interruption (fig. 3.14). If you execute the options individually to display more detail, you must preserve the sequence of the steps. Use option 6 to review the nodal displacements before proceeding to POSTPROCESS.

If you only need a plot of the nodal displacement components, you may proceed immediately from the principal menu (5.05) to PLOT. Frequently, however, you need to find various stresses and strains. For these, proceed to POSTPROCESS.

POSTPROCESS

In POSTPROCESS (fig. 3.15 and 3.16) the principal menu (5.05) indicates that this program segment, like SOLVE, is computation bound and requires no additional user-input. Use option 1 to find the element and nodal stresses and strains. You can ONLY list each of these classes of computed results here. You will ONLY be able to produce plots in the final program segment PLOT.

Fig. 3.15 POSTPROCESS abstract

```
        APPLIED FINITE ELEMENT ANALYSIS:
           AN APPLE ][ IMPLEMENTATION

               COPYRIGHT, 1985

              POSTPROCESS (5.00)

ABSTRACT:
NODE DISPLACEMENTS DETERMINED IN 'SOLVE'
ARE USED TO DEFINE ELEMENT STRAINS,
ELEMENT STRESSES, AND NODE STRESSES.
STRESSES ARE DEFINED IN COORDINATE AND
PRINCIPAL DIRECTIONS.

IN ADDITION NODE REACTIONS AND NODE NEW
POSITIONS ARE DETERMINED.
```

Fig. 3.16 POSTPROCESS principal menu

```
POST-PROCESSING (5.05)

 -- SOLVE ALL EQUATIONS --
1. DO ALL CALCULATIONS (MENU OPTIONS
   2 - 7)

 -- SOLVE FOR STRAINS AND STRESSES --
2. ELEMENT STRAINS & STRESSES
3. ELEMENT PRINCIPAL STRESSES
4. NODE COORDINATE STRESSES
5. NODE PRINCIPAL STRESSES

 -- OTHER CALCULATIONS --
6. NODE NEW POSITIONS
7. NODE REACTIONS

 -- LISTING --
8. LIST OUTPUT


0. NONE OF THESE
```

PLOT

PLOT produces plots of the mesh, the boundary conditions, and the computed results. (See fig. 3.17 and 3.18.) To increase resolution you may draw enlargements. You can either enlarge an arbitrary user-specified rectangular portion or produce several multiple plots using a common enlargement in order to produce a large composite picture. An elaborate labelling capability allows you to produce finished plots.

Fig. 3.17 PLOT abstract

```
        APPLIED FINITE ELEMENT ANALYSIS:
           AN APPLE ][ IMPLEMENTATION

                 COPYRIGHT, 1985

                   PLOT  (6.00)

ABSTRACT:

PREPARE FINISHED PLOTS OF GRID, BOUNDARY
CONDITIONS, OR RESULTS.  ANY PLOT MAY BE
LABELLED AND PRINTED.  ZONE PLOTS MAY BE
SELECTED TO ENLARGE PARTS OF INTEREST.
```

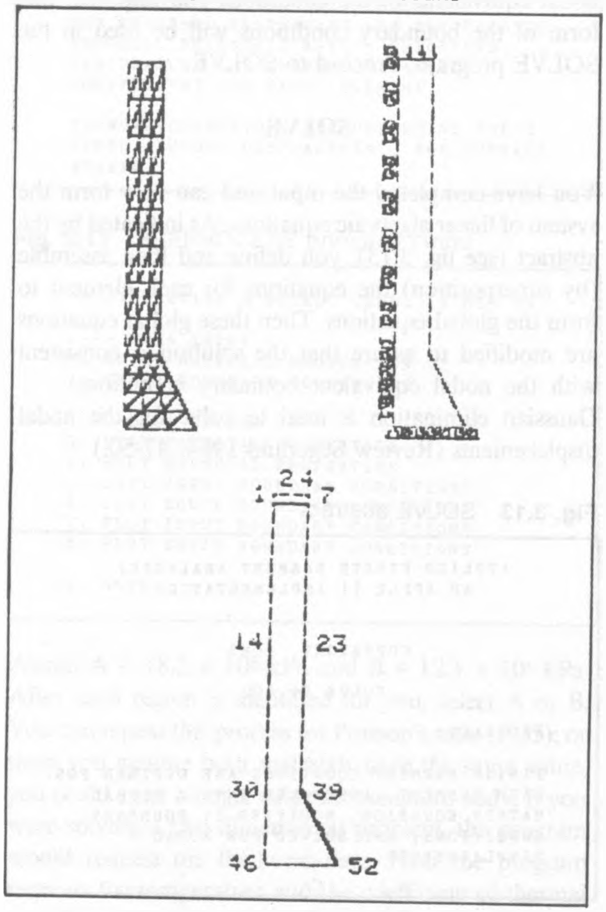
Fig. 3.18 PLOT principal menu

```
          PLOT  (6.05)

     -- PLOT FORMULATION --
  1. PLOT GENERATED GRID
  2. PLOT BOUNDARY CONDITIONS

     -- PLOT RESULTS --
  3. PLOT NODE DISPLACEMENTS
  4. PLOT ELEMENT STRAINS
  5. PLOT ELEMENT STRESSES
  6. PLOT NODE STRESSES

     -- OTHER --
  7. RETRIEVE PICTURE FROM DISK

  0. NONE OF THESE
```

Using the conventions followed throughout the program, you may exercise considerable control over label placement. Use the left and right arrows to move the cursor. Press 'S' to select a label to be changed. An 'X' adds or removes the label. After a label has been selected by 'S' use the traditional I, J, K, and M diamond to move the label in large increments. Smaller increments of movement are possible using control-I, control-J, control-K, and control-M. Use 'Q' to quit the modification of the select label. Two successive 'Q''s allow you to exit the label edit routine. A 're-draw' option allows you to erase the label background and destructively (DRAW rather than XDRAW) re-write the label.

Use the principal menu options 1-6 (fig. 3.18) in any order to display the previously generated data files. (If you are solving a very large problem, you may need to eliminate unnecessary arrays by doing a cold restart (exit and RUN) and then proceeding directly to a specific plot.) Figure 3.19 illustrates various types of plots. Use option 7 to retrieve an existing picture file saved earlier in options 1-6.

Fig. 3.19 Axisymmetric column results



3.2 Illustrative Examples

Tapered Cantilever

This next example is also adapted from Gustafson (1977, 24) and provides additional validation for this implementation. Figure 3.20 characterizes the problem. The choice of mesh generation regions (figure 3.21) is obvious and the program quickly produces a mesh.

The left end of the cantilever is constrained. The load on the cantilever is vertically downward (fig. 3.22). Observe that the equivalent nodal force is not distributed equally among the nodes. Results are shown in figure 3.23. Before leaving this example, we note that the linear triangular element models bending less satisfactorily than it does a tension load.

Fig. 3.20 Tapered cantilever beam (Gustafson 1977, 24)

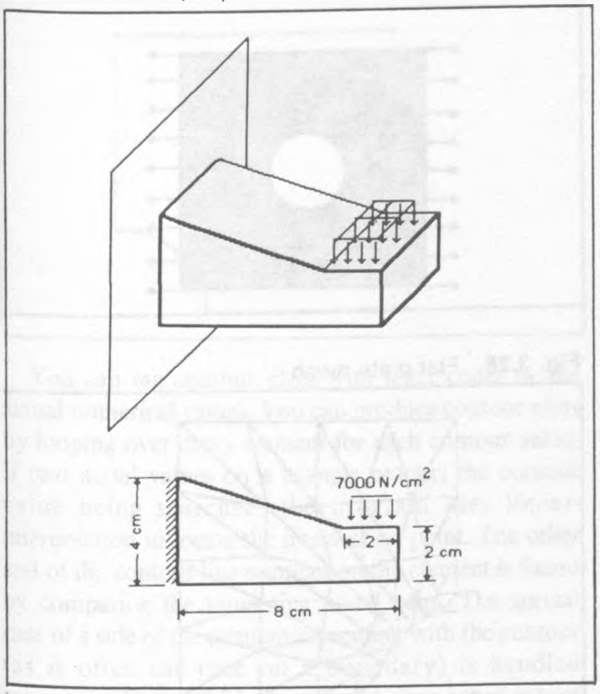


Fig. 3.21 Regions and elements for cantilever

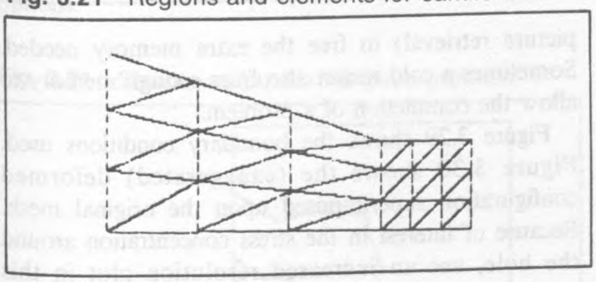


Fig. 3.22 Boundary conditions for cantilever

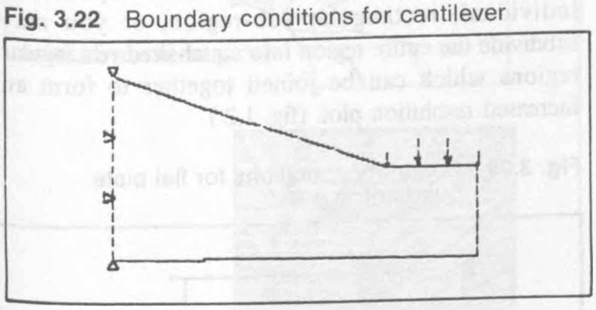
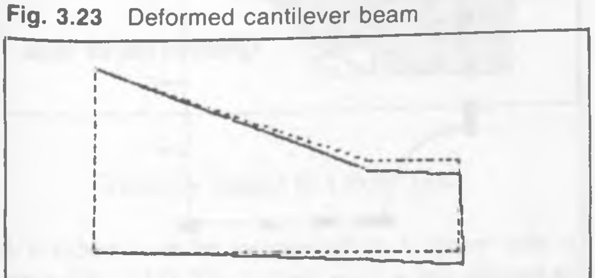


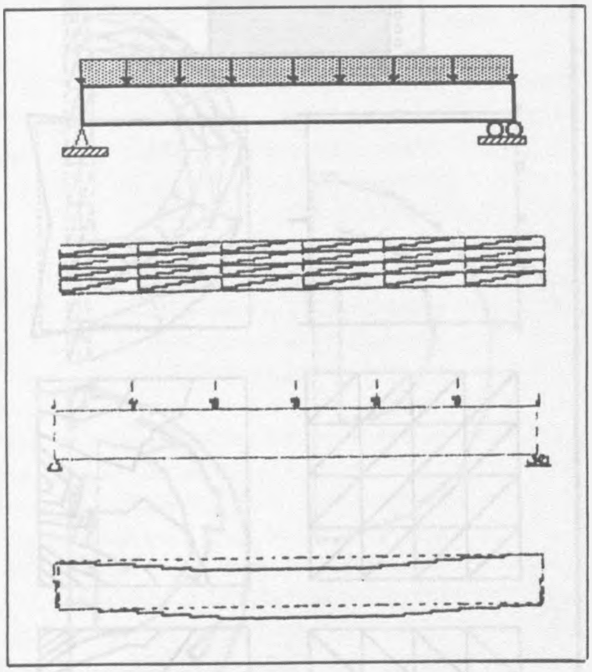
Fig. 3.23 Deformed cantilever beam



Simple Beam with Uniform Loading

Figure 3.24 presents an analysis of a simple beam with uniform loading.

Fig. 3.24 Uniformly loaded simple beam



Symmetry Test

This example illustrates the importance of generating a mesh which preserves the inherent symmetrical conditions in a problem (fig. 3.25). The symmetrical mesh at the bottom produced using diagonal reversal in GRID (option 2) gives better results than the asymmetrical mesh.

Patch Test

The simple example in figure 3.26 demonstrates that the element performs properly.

Tensile Loading of a Thin Flat Plate with a Hole

The classical problem of the tensile loading of a thin flat plate with a hole (Timoshenko and Goodier 1951; Peterson 1954; Roark and Young 1975; Anderson 1983) is presented in figure 3.27.

Figure 3.28 depicts a quadrant of the plate subject to loading. Two eight-node mesh generating regions were used to generate the mesh. The refined mesh on the right approaches the maximum capacity of the program. We deleted segments of PLOT (zoom and

Fig. 3.25 Effect of mesh symmetry

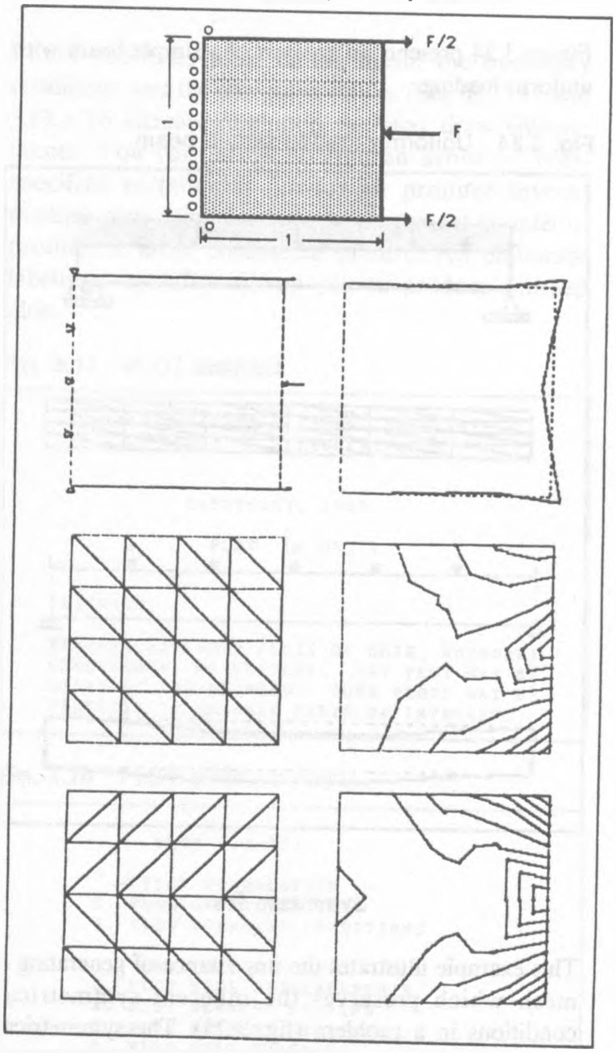


Fig. 3.26 Patch test

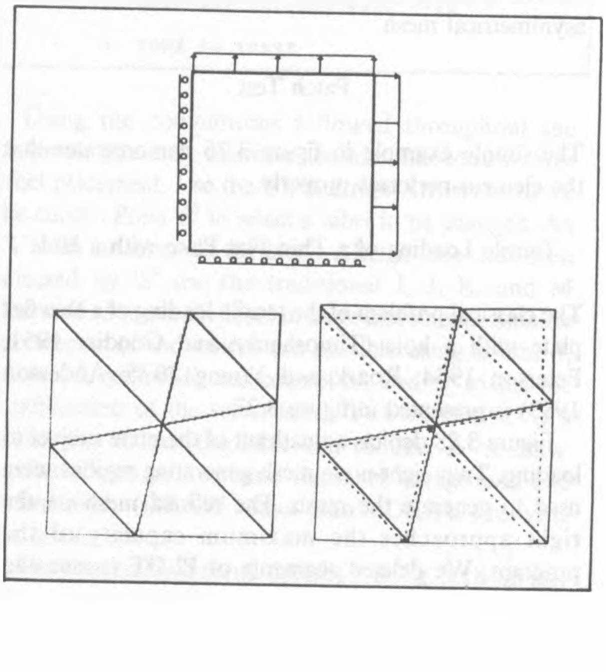


Fig. 3.27 Tensile loading of a thin flat plate with a hole

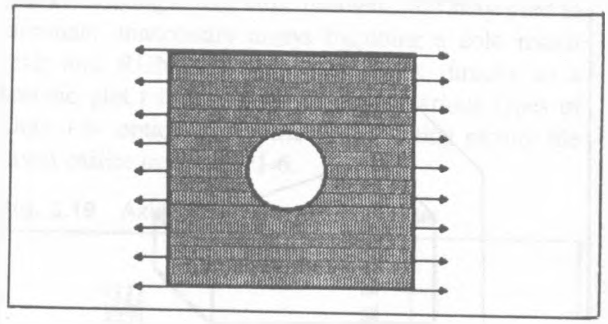
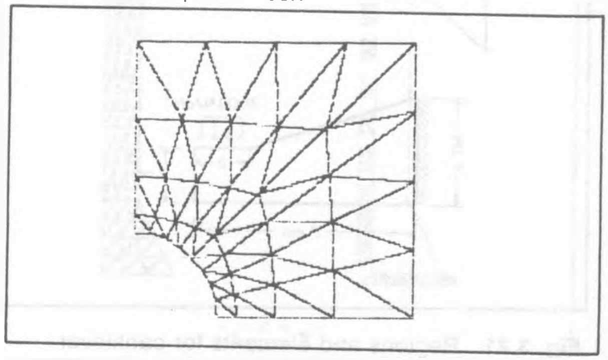


Fig. 3.28 Flat plate mesh



picture retrieval) to free the extra memory needed. Sometimes a cold restart also frees enough memory to allow the completion of a problem.

Figure 3.29 shows the boundary conditions used. Figure 3.30 shows the (exaggerated) deformed configuration superimposed upon the original mesh. Because of interest in the stress concentration around the hole, use an increased resolution plot in this example. You have two possibilities. You may select an individual, rectangular sub-region, or you may subdivide the entire region into equal-sized rectangular regions which can be joined together to form an increased resolution plot. (fig. 1.2.)

Fig. 3.29 Boundary conditions for flat plate quadrant

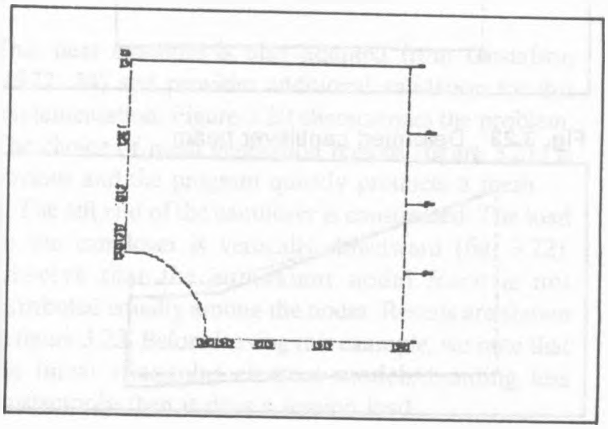
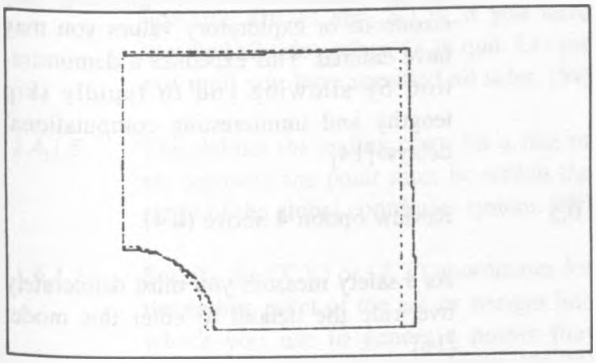
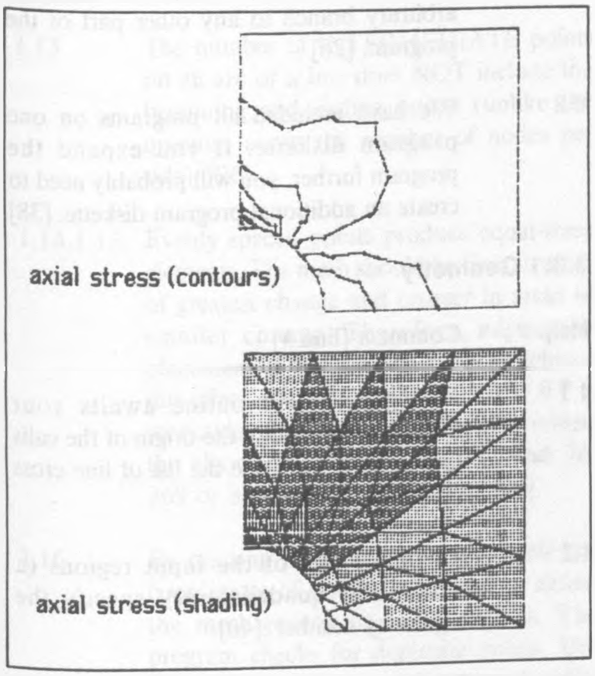


Fig. 3.30 Deformed plate



You can tag contour plots with letter codes or the actual numerical values. You can produce contour plots by looping over every element for each contour value. If two nodal values on a triangle bracket the contour value being searched, the program uses linear-interpolation to locate the intersection point. The other end of the contour line segment on the element is found by comparing the remaining nodal pairs. The special case of a side of the element coinciding with the contour (as is often the case on a boundary) is handled separately. Figure 3.31 illustrates various other output options.

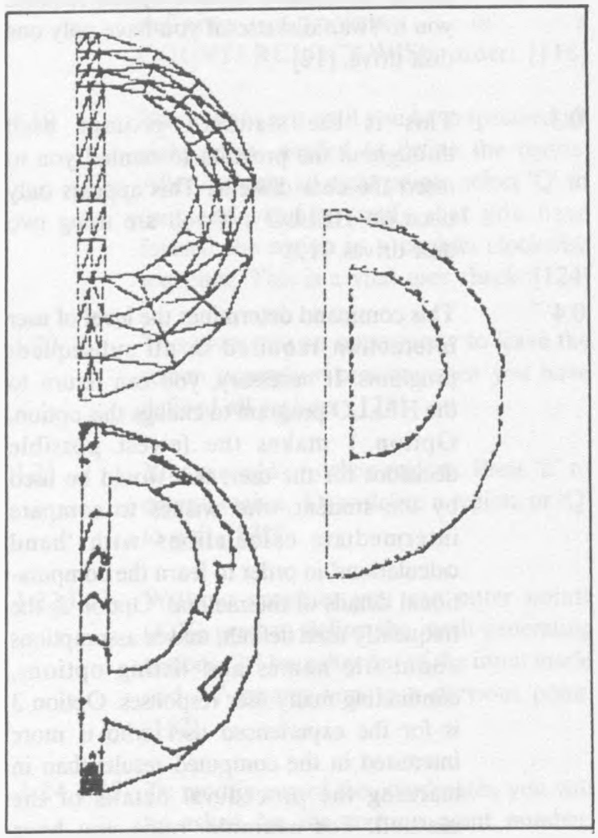
Fig. 3.31 Sample output for flat plate problem



Cranberry Subject to a Point Load

A cranberry can be represented as a sphere with a cavity (fig. 3.32). The bottom point is not allowed to move in either the R or Z directions. Rollers along the axis of symmetry indicate that no motion is allowed in the R direction. Figure 3.32 also shows the results.

Fig. 3.32 Point loading for a cranberry



3.3 Help Messages for Elasticity

At your discretion during program execution, you can obtain additional clarification of a question by typing a question mark as the first character immediately after the bracketed default response and pressing 'return'. The integer part of the help number identifies the program segment being executed; the fractional part identifies the specific input request. Press 'return' to return to input mode. Note: Help messages for elasticity and heat conduction are similar but different. Use the appropriate version.

3.3.0 Hello

Help #	Comment [line number]
0.1	Use uppercase letters throughout. If you are using an Apple II-e or Apple II-c, remember to set the 'caps lock' key. [10]

0.2 A second disk drive is highly desirable. Otherwise, insert the program diskette whenever you are changing programs and then replace it with the data diskette. In PLOT the picture retrieve option requires you to swap diskettes if you have only one disk drive. [12]

0.3 This is the standard prompt used throughout the program to remind you to insert the data diskette. This appears only once (in HELLO) if you are using two disk drives. [12]

0.4 This command determines the level of user interaction required in all subsequent programs. If necessary, you can return to the HELLO program to change this option. Option 1 makes the fewest possible decisions for the user; this would be used by the student who wishes to compare intermediate calculations with 'hand calculations' in order to learn the computational details of the method. Option 2, the frequently used default, makes assumptions about file names and listing options, eliminating many user responses. Option 3 is for the experienced user who is more interested in the computed results than in learning the procedural details of the method. For example, once you have formulated the problem , the calculations in SOLVE and POSTPROCESS proceed without user attention. Option 4 is for the classroom instructor who might accidentally erase data files during a lecture demonstration. This option suppresses the saving of any data, permitting the data diskette to be used with a 'write protect' tab.

Normally, the program disables any pre-existing data files which might become inconsistent with the immediately entered data. For example, if you alter the coordinates of a node, then you automatically disable but do not erase any pre-existing solution for the old coordinates.

To use option 4 you must first create a demonstration problem. In a demonstration, you retrieve these existing data files. At each stage you can explore the various options in detail and retrieve the next phase of the demonstration, ignoring any erroneous or exploratory values you may have entered. This expedites a demonstration by allowing you to rapidly skip lengthy and uninteresting computational delays. [14]

0.5 Review option 4 above (0.4).

As a safety measure you must deliberately over-ride the default to enter this mode. [16]

0.6 This is the main title page for the program. [20]

0.7 This is referred to as the MAIN MENU. Normal program execution proceeds sequentially down the list of options, and the default menu item is automatically updated when you finish a segment. Each program segment has an exit option which allows you to proceed to the next step (item #1), return to this main menu (#2), remain in the then current segment (#3), or exit the entire program (#0). A return to this main menu allows you to make an arbitrary branch to any other part of the program. [28]

0.8 We have included all programs on one program diskette. If you expand the program further, you will probably need to create an additional program diskette. [38]

3.3.1 Geometry

Help # Comment [line #]

1.1 This general subroutine awaits your response. To identify the origin of the calls to this subroutine, see the list of line cross references. [4]

1.2 To delete one of the input regions (a curvilinear quadrilateral), specify the region by number. [40]

1.3 Each curvilinear quadrilateral used to generate a mesh of triangular elements can have from 2 to 9 element nodes on each side (including the two vertices on that side of the quadrilateral but not including the mid-side point). After specifying the number for this side, either select another side (use arrows and 'S') or, if you have specified all sides, select 'Q' to quit. Do not exit until you have specified all sides. [54]

1.4,1.5 This defines the ending point for a line or arc segment; the point must be within the range of the global coordinate system. [60]

1.6,1.7 Specify the (X,Y) or (R,Z) coordinates for the ending point of the arc or straight line which you use to generate points that define quadrilateral input region(s). [72,74]

1.8,1.9 A circular arc generates the points needed to define a quadrilateral input region. Specify the center of the arc. [80,82]

1.10,1.11 Specify the (X,Y) or (R,Z) coordinates for the beginning point of an arc or straight line which you use to define quadrilateral input regions. [86,88]

1.12 An arc generates points to define a quadrilateral input region. The signed angle is in degrees; counterclockwise is positive. [92]

1.13 The number of INTERMEDIATE points on an arc or a line does NOT include the beginning and ending points (unlike the question about the number of nodes per side). [96]

1.14,1.15 Evenly spaced points produce equal-sized elements. The mesh should be finer in areas of greatest change and coarser in areas of smaller change. Therefore, adjust the placement of the mid-side nodes to achieve this effect. Usually a value of 0.3 to 0.7 is workable. If placed too near the extremities, the algorithm will fail (see Segerlind 76, 369 or Steinmueller 1974). [96,100]

1.16 By pressing 'E' or 'D' enter or delete the coordinates of points to be used to define the mesh-generating input regions. The program checks for duplicate points. Use 'Q' or '>' to quit or use '<' to branch to the menu (1.05). [108]

1.17 When you have selected 'E' above (1.16), you can enter a point, a straight line, or an arc; quit; or branch back one question. [110]

1.18 To begin creating a region you must specify its number. Then you 'point' to the eight defining points in a COUNTERCLOCKWISE order. [118]

1.19 You cannot exit until you have specified all eight points needed to define the region. After defining all eight points, select 'Q' to quit and visually verify that you have formed the region in a counter clockwise sequence. This is a vital user check. [124]

1.20 This gives you an opportunity to leave the region generation process when you have defined all regions. [128]

1.21 You can add or delete regions. Press 'E' to enter a region, 'D' to delete a region, or 'Q' to quit. [130]

1.22,1.23 Without graphics you can enter points (X,Y) used to define the mesh generating regions. '>' branches out of the input mode and '<' returns you to a previous point. [152]

1.24,1.25 To modify any of the coordinates, you will be asked for the starting point number. [154]

1.26,1.27 Select 8 points from the previously entered list of coordinate pairs (without graphics) to form a region in counterclockwise order. [158,160]

1.28,1.29 Here, as throughout the program, you can revise your previous input. [162]

1.30,1.31 Specify the number of nodes for each pair of opposite sides. Without graphics, you must remember to match the values for regions having a common boundary. [166]

1.32,1.33 As throughout, you have an opportunity to modify previously entered data. [168]

1.34 The edit menu allows you to revise any of the preceding steps (input coordinates, selection of region points, and designation of the number of element nodes to be placed on each side of the mesh generating region). [172]

1.35 Here you can edit the mesh generating regions. [178]

1.36 When you save a data file, specify the file name to be used. Unless you will have several alternative problem formulations, accept the default. The free-run mode uses the default names automatically without additional user action. [226]

1.37 When retrieving a data file, specify the file name. The free-run mode uses the default name. [234]

1.38,139 Error traps allow you to correct disk problems. [258,260]

1.40 This is the only opportunity for you to choose between a two-dimensional and an axisymmetrical geometry. The element cross-section appears to be triangular in either case; however, in the axisymmetrical case the triangle sweeps out an annular region. Mathematically, the problems are similar but different (Segerlind 1984, 87-99 and 165-176). Likewise, the plots produced are similar in appearance but different. For axisymmetric problems, you must use the Z axis as the axis of symmetry (see Segerlind 1984, 166 fig. 13.1). [WARNING: This question is only asked during a cold start so do NOT use the control Q warm restart command here, bypassing the answer to this question.] [274]

1.41 This program establishes coordinate points that are used to define quadrilateral regions used in the following GRID program to generate the discretization mesh. You need to use judgment creating this mesh because this step influences the accuracy of the solution.

Number the principal menu for each program (X.05), as here. Because data input with graphical support is both easier and less prone to undetected errors, you will usually elect options 1 and 3 in that order. Nevertheless, data input with a closer correspondence to the batch programs of Segerlind is possible with options 2 and 4.

Here, as in subsequent programs, you can review the generated data that is stored immediately on disk. Both a tabular listing and a graphical representation are possible. During listings, the output will be presented by the screenful. Press 'return' for the next screen or press 'space' for the next line or press a number, 0-9, to scroll the data. [284]

1.42 Each program segment has an exit menu. The first option leads you to the next logical step in the solution and is the default that can be accepted by simply pressing 'return'. Should you need to branch to a different program segment or to change the user interaction default, the menu allows you to return to the HELLO program (see Help 0.4).

Throughout the program we have attempted to give you an opportunity for second thoughts. Here, that means you can return to the current program. You can also use the 'less than' symbol, '<', with 'return' to go back to a previous step if you type it as the first character after the bracketed default prompt.

To keep you from entering the middle of a FOR-NEXT loop, etc. the 'branch backward' option sometimes sends you back further than you might have expected. Sometimes you will find it even harder to anticipate the action of the 'branch forward' command '>' used when you want to skip a section. (Note: The variable BC% is used consistently throughout the source code to route the branching.)

We have placed the termination option last on each menu throughout and have numbered it '0'. [286]

1.43 If you are reviewing a problem in demo mode (Help 0.4) or are revising a previous solution, retrieve the coordinate data (see Help 1.41 - the data being retrieved was created with option 1 or 2 of menu 1.05). If retrieved, the defaults presented reflect the retrieved data; a sequence of 'returns' allows you to quickly advance to the position you wish to alter.

To change the number of nodes on a side of the mesh generating region, a frequent modification, go immediately to option 3 or 4, skipping options 1 and 2 of menu 1.05. [294]

1.44 Here you can retrieve the region data from a previous exercise (see Help 1.43). This data includes the connectivity of the 8 points defining each region and the number of element nodes to be placed on each side (including the vertices). [296]

1.45-1.48 To make a global plot of the problem including all regions, specify the limits for the coordinate axes. You exclude coordinates of points outside this region. [298,300]

3.3.2 Grid

Help #	Comment [line number]
2.1	This general purpose subroutine allows you to decide when to proceed. Refer to the list of line cross-references to determine the origin of calls to this subroutine. [4]
2.2,2.3	This code allows you to relocate a node. Enter the new (X,Y) or (R,Z) coordinates. [18]
2.4,2.5	Use the graphics cursor to point to individual nodes and elements. Begin the sequential review at any node or element. Use the left and right arrows to move through the list and press 'Q' to quit. After leaving node identification, begin element identification using the same commands. [190,194]
2.6	If you wish to retain only one boundary value problem on the diskette (the usual case), use the default filenames by pressing 'return'. If you wish to retain several variations in the problem, to prevent the overwriting of data, you must use a different file name in saving the data. [Note: you must remember any alternate names for subsequent retrieval (see Help 2.7).] [204]
2.7	Here as in Help 2.6, specify the name of a data file for retrieval. The free-run mode (see user interaction level Help 0.4) automatically selects the defaults for Help 2.6 and 2.7. [214]
2.8,2.9	These are two error recovery options. You can unlock a data file (necessary if you wish to replace the data using the same file name) or you can review the existing list of file names using the catalog command. [234,236]
2.10	This is the principal grid definition menu (2.05). Characteristically, you proceed sequentially through the list. Option 1 allows you to handle the process with fewer user responses, provided you wish to execute options 2-4. If you are exploring the computational details, you may wish to proceed more deliberately. Consistently throughout the programs, you have an opportunity to immediately review the results (options 5-7). Select option 0 if you are ready to leave this program (i.e., go to the exit menu). [258]
2.11	The exit menu has the standard options - proceed to the next logical program segment, return to the menu in the HELLO program for arbitrary branching, return to the current program, or leave the program entirely. The single-disk-drive user must swap diskettes. [262]
2.12	This option appears only if you are not in free-run mode and allows you to decide the level of interaction required in this program. If you are studying the methodology, you may want access to more computational details. [270]
2.13	This is menu option 2 and generates the mesh nodes and elements. Here you can (1) generate the grid automatically using the data entered in the GEOMETRY program or (2) enter the data without graphics support (necessary if you are creating specialized meshes such as Segerlind's

figure 11.8 figure 16.4 (1984, 149)), or (3) generate the cover page which cannot be generated automatically. Usually you can generate satisfactory meshes automatically. The third option is useful if you can generate the basic mesh automatically and then make minor adjustments. Make the modifications to elements and nodes at the end of the list. Note that node relocation and diagonal reversal is handled easily elsewhere. (See Help 2.33.) [276]

2.14 The display option allows you to choose to utilize plotting. Under the free-run option (global or in Help 2.12) this question defaults automatically to 'without user interaction'. [278]

2.15 The grid listing option also allows you to examine the details of the calculation process. Option 1 is usually preferred, deferring the listing to options 5-7 of menu 2.05 (see Help 2.10). [280]

2.16 In the free-run mode use the grid plotting option default of plotting without labels. If you label every node or element, the labels can become cluttered. After you have completed the solution, you have an opportunity to label selectively in PLOT. The extensive use of graphics greatly diminishes your need to be aware of the element and node numbers. Usually you need to know only a few strategic node numbers. You can easily move a significant portion of the procedural details into the background. [282]

2.17 This is another option that you can use only when you are not in the free-run mode. Select option 2 only if you wish to explore the behavior of the renumbering algorithm.

By judiciously renumbering nodes, you can greatly expand the scope of problems that you can solve using this microcomputer. While execution of this part of the program can become significant, the real benefit is a more compact formulation of the global equations needed if significant instructional problems are to fit within the limited memory. The additional computation at this step also reduces execution time. The bandwidth reported here represents the maximum difference between the node numbers connected through a side of a triangular element plus one. This corresponds to the width of the array required to store the stiffness matrix if each node has one degree of freedom. Here each node has two degrees of freedom so the problem bandwidth is twice this number. Since the system of simultaneous linear algebraic equations is resident in memory during the Gauss elimination solution, the node renumbering process becomes a significant step. (Note 1: other solution techniques permit a solution to be found if only parts of the large array are resident at any time. Note 2: the Collins renumbering algorithm does not guarantee optimal renumbering. The original renumbering is retained if the bandwidth is not reduced.) [290]

2.18 In the initially generated mesh, each of the three sides of an element is plotted even when a side coincides with a previously plotted element. To avoid redundant double plotting of common element boundaries and to identify the boundaries required in the PREPROCESS program, unique lines are identified. [296]

2.19 You can elect to list either the original or the revised node numbering. This is the first question in menu option 5. [318]

2.20,2.21 When you elect menu option 7 to plot the mesh and identify elements and nodes, you can select the original or renumbered nodes.

2.22,2.23 When you wish to modify the mesh by moving nodes, enter the new coordinates here. [320]

2.24,2.25 To check the counterclockwise ordering of nodes in an element, specify a region. [322]

2.26 When you enter nodes and elements without graphics, you can retrieve the existing data file that can be treated as the default. This file may actually have been created using graphics. [336]

2.27,2.28 Because the program makes more efficient use of memory through dynamic dimensioning of arrays (i.e., arrays may be created and erased during execution), you need to specify the array size for just the memory required. If you are modifying existing data, note that the changes occur at the end of the list. Specifically, if you are reducing the array size, the existing array is truncated and copied into the new array. If you enlarge the array, append the new additions to the end. If you use the mesh generator to create the bulk of the mesh to be modified here, define the other regions such that you make changes in them at the end of the list. [338]

2.29 You can enter the coordinates of nodes, the global numbering of the nodes, or neither. Observe the COUNTERCLOCKWISE enumeration of the nodes when you enter the global node numbers. [342]

2.30,2.31 The editing process permits you to enter values.

2.32 Use '<' to decrement the counter and '>' to branch out (i.e., end the editing). Start at a user specified index and specify the maximum subscript for the nodal coordinate array or array of node numbers for the save. [346,348]

2.33 You can elect to either adjust node positions or reverse diagonals. 'Well-shaped' elements give better computational results; you should avoid triangular slivers. If you move a node, the program erases and redraws the old elements. The mesh generation routine automatically selects the diagonal of an element pair that gives the better aspect ratio. Sometimes, however, you may wish to over-ride this (e.g., to obtain greater symmetry in the mesh or to avoid having two sides of an element fall in the corner of a boundary). [352]

3.3.3 Preprocess

Help # Comment [line number]

3.1 This general-use subroutine allows you to receive information and awaits your response before clearing the screen and continuing. See the line cross-reference listing to locate the calls to this subroutine. [4]

3.2 This optionally displays the list of files saved on the data diskette. [44]

3.3 If you are to take into account stress due to thermal expansion, you must supply the reference (or no thermal stress) temperature. [74]

3.4 You can base data entry on three different categories - the entire body, the mesh generating region, or individual elements. When the property is a single constant throughout the entire body, enter that single value by selecting option 1. Then if you can characterize input (mesh generating) regions by a constant for each region, select option 2. Finally, if the properties vary from element to element, select option 3. Note: if the property is relatively uniform for the entire body but differs in a few places, choose option 1 and use option 2 or 3 to edit the initialized exceptions. [76]

3.5,3.6 Enter the uniform property value here. Enter Poisson's ratio separately; the value is bounded between 0 and 0.5. [80,82]

3.7-3.9 If a property can assume only a few values (up to 6), assign those values to letters and circumvent the repeated entry of multidigit numbers. When assigning the properties to the problem (see Help 3.10), either enter the letter for a coded value or over-ride the code by entering a numerical value. First specify the number of coded values and then supply the actual coding. [86,90,92]

3.10 Input the properties for each element. Enter a code, the actual number, or 'Q' to quit. [98]

3.11-3.13 As is customary, you have an opportunity to revise your input. Edit by region or by element. Specify the beginning region or element number. [104,106]

3.14 Select the property to be edited. For axisymmetric problems the menu includes the thickness property but will not allow you to select that option. [112]

3.15 Enter properties by region in the same manner that you enter properties by element (see Help 3.9). [122]

3.16 Enter or delete boundary conditions for a single node or for a range of nodes with a single type of boundary condition. Specify (1) the boundary condition type (nodal point force, displacement, or surface stress (or free boundary if not specified)); (2) the direction of the boundary condition component (X,Y or R,Z directions, the normal to the boundary with the outward direction taken as positive or the tangent to the boundary with COUNTERCLOCKWISE taken as positive); (3) the value associated with each component; and (4) either a single node or a range of consecutive nodes around the boundary. When you specify a range of nodes, use a COUNTERCLOCKWISE traversal of the boundary to determine which is the beginning and the ending point. Use 'E' to enter a boundary condition and 'D' to delete. Use 'Q' to quit. The branching codes '<' and '>' take you back to the instructions or forward just as 'Q' does. [134]

3.17 Now select the boundary condition type to be entered or deleted. The three possibilities are (1) nodal point force, (2) displacement, and (3) surface stress. '<', '>', and 'Q' all branch to the previous question. [138]

3.18 Surface stress may occur on the boundary that you see edgewise on the screen. Give the surface stress for the X and Y or R and Z directions, for the normal to the surface with the outward direction taken as positive, or for the tangent to the surface with counterclockwise taken as positive. [142]

3.19,3.20 Point force and displacement boundary conditions have two possible modes of application: (1) X-direction components and (2) Y-direction components. Specify the force or displacement value. The '<' branches to the logically previous question to allow you to select a different boundary condition type; '>' branches one step further back (Help 3.16) to allow you to decide to enter, delete, or leave this option. [146,148]

3.21 Specify boundary conditions at a single node or along the boundary over a range of nodes. Use the left and right arrow keys to move the cursor to the desired starting and ending point; press 'S' to select. The program subsequently converts all boundary conditions into nodal equivalents before modifying the global system of equations. Note: You must observe the counterclockwise convention for the boundary starting and ending points. Should you forget, the program will apply the boundary condition to all of the remaining boundary. To recover, delete the condition or use control-Q to do a warm start to menu 3.05. Select option 2 and begin entering the boundary conditions again. The program uses the XDRAW command to place the symbols on the screen. This means that if you enter a condition at a node twice, the symbol will disappear and can be misleading. This makes the warm restart approach somewhat safer. [152]

3.22 For multiply-connected regions (i.e., regions having more than one boundary), you are asked whether you wish to specify boundary conditions on another boundary. [154]

3.23 If not in free-run mode, you have an opportunity to have 'second thoughts' and revise any of the boundary conditions. The program saves the boundary condition data (type codes and values) and then calculates and saves the equivalent nodal boundary conditions. [156]

3.24 Supply the name for the data being saved. The number of underscores indicates the maximum name length. Space characters are allowed in the file name. Usually you simply press 'return' to accept the default. The free-run mode does this automatically. Should you wish to place several problems on the diskette by renaming

'FILEINFO.TXT' and preparing another FILEINFO.TXT WITHOUT erasing the diskette, add the problem keyword to the beginning of the file to identify the connection. [198]

3.25 You need to enter the file name of the data you wish to retrieve. Press 'return' to accept the default; free-run mode does this automatically (see Help 3.24). [206]

3.26 If a data file happens to have been locked when you try to save another file with the same name, you have an opportunity to unlock the file here. [224]

3.27 If the program cannot locate the requested data file, you have an opportunity to swap diskettes. [226]

3.28 When using a single disk drive, you must exchange the program and data diskettes. You need the program diskette to load the program, but the data diskette must be resident during execution. [236]

3.29 This is the principal PREPROCESS menu (3.05). As usual, progress sequentially through the menu options 1 and 2. The remaining options allow you to list and plot the properties and the boundary conditions. The program converts the user-specified boundary conditions into equivalent nodal values that you will use later to assure that the boundary conditions are satisfied in the solution.

Remember that the tabular data is listed by the screenful. If you need more than one screen, press 'return' to obtain the next screen or press 'space' to obtain the next line or press a number, 0-9, to scroll the screen continuously until you stop it by pressing the space bar. You can change the scroll speed 'on-the-fly'. [258]

3.30 The exit menu (3.9) presents the standard exit options - proceed to the next program, return to the main menu in the HELLO program, remain in this program, or terminate the calculations. In addition, the program provides a link to a text file creation program. You can customize that program, given only in outline form in this book, to generate an input file for mainframe computer execution. In other words, utilizing Segerlind's programs or the extensions developed by Robert J. Gustafson, you can use the microcomputer with its low-cost graphics to formulate problems that you want to execute on a larger computer (see references).

The exit menu option 3 permits you to use a separate program diskette for the user-supplied program. This approach permits you to write such a program without requiring a detailed knowledge of the tightly compressed PREPROCESS program.

If you do not need the full generality of PREPROCESS (e.g., the element by element variability), then you only need to store integer codes for material property sets, saving enough memory to permit you to formulate much larger problems. [262]

3.31 If you wish to create a text file or are using a single drive, you must insert the correct program diskette. The program diskette contains insufficient space to include the 'create text' file program. [266]

3.32 If you are revising a previously formulated problem in demo mode or simply changing the size of the problem, answer 'Y' to retrieve the earlier material properties data. [274]

3.33 You have reached this point by completing the input of the material properties data. A 'Y' answer sends you directly to the boundary conditions input; an 'N' returns you to the menu 3.05 (see Help 3.29). [278]

3.34 Retrieve the currently active boundary condition files. If present, these data become the default prompts for the boundary condition re-formulation. If the data are not present but have been designated as inactive in the FILEINFO.TXT file, you are advised of the problem. [280]

3.35 For review you can plot the material properties generated by this program. [300]

3.36 When you use shading to represent the magnitude of a calculated value, the density of the dots illustrates the range from high to low. If you are using a color monitor, you may need to adjust the shading combinations set in line 246. [310]

3.37 Since you are describing a uniform value, you have that information in both graphical and text formats. [312]

3.3.4 Solve

Help #	Comment [line #]
4.1	This general purpose subroutine allows you to decide when to proceed. Refer to the list of line cross-references to determine the origin of calls. [4]
4.2,4.3	During the extended computations of this program you can change the level of display details. To reach this point you pressed 'esc'; if you now press 'D' you can choose to display (1) only the final global matrices, (2) element and global matrices, (3) details of the calculations, or (4) no display. [6,8]
4.4	Assemble the global force and stiffness matrices before you constrain any of the degrees of freedom by boundary conditions. [74]
4.5	While applying the force boundary conditions, you can display (1) the initial global stiffness matrix, (2) the updated global stiffness matrix, (3) both, or (4) neither. [80]
4.6	Combine the boundary forces and initial thermal forces before you update the global force and stiffness matrices to satisfy the displacement boundary conditions. [96]
4.7	While applying the displacement boundary conditions, you can display (1) the global stiffness matrix already modified for boundary forces, (2) the global stiffness matrix updated for displacement boundary conditions, (3) the force (right hand side) vector, (4) the displacement updated force vector, (5) all of the above, (6) 2 and 4 above, or 7) none of the above. [112]
4.8	Apply displacement boundary conditions before you attempt to solve the system of equations for the nodal displacement. [130]
4.9	You can display the nodal displacements as you compute them provided you are not in the free run mode. [138]
4.10	If you are not using the free-run mode, you can select the name for a data file. Normally you accept the default unless you wish to save several versions of the same data file. [174]
4.11	If you are not in free-run, select the specific version of the data file you wish to retrieve. The free-run mode always uses the default. [184]
4.12,4.13	These options allow you to recover from trying to save a data file having the same name as a locked file or to determine the name of a file you wish to retrieve. [206,210]
4.14	Specify whether you wish to assume plane stress or plane strain deformation. For a thin body with no loads perpendicular to the plane of the body, assume plane stress. On the other hand, if the dimension of the body normal to the applied load may not change, assume plane strain. See Chapter 23 of Segerlind for further details. [228]
4.15	This is the principal SOLVE menu (4.05). Choose 1 to proceed through options 2-5 without user interaction. Otherwise, sequentially assemble the global force and stiffness matrices and update these matrices for any force boundary conditions and then for displacement boundary conditions. Finally, solve the system of equations for the nodal displacements. As is the custom, you can examine the results before leaving this program. [236]
4.16	The exit menu (4.7) routes you to POSTPROCESS, routes you back to the

HELLO program, allows you to remain in SOLVE, or allows you to terminate the program. If you only need to plot the mesh, the boundary conditions, and the nodal displacements, go immediately to PLOT via HELLO. [238]

4.17 Option 1 coordinates the execution of the other calculations. If you have not elected a global free-run while in the HELLO program, you can choose to explore the computational details of the assembly and solution. [246]

3.3.5 Postprocess

Help #	Comment [line #]
5.1	This general purpose subroutine allows you to decide when to proceed. Refer to the line cross-reference list to determine the origin of calls to this routine. [4]
5.2	When solving for the element strains and stresses (using the nodal displacements), you can elect to display the results as obtained. [30]
5.3	When solving for the element and nodal principal stresses, you can display the principal stresses or suppress the display. Option 8 of the principal menu 5.05 (see Help 5.12) presents another opportunity for listing the results. [56]
5.4	Form the conjugate principal stress matrices and solve using Gaussian elimination calculations. [74]
5.5	Do you wish to display the nodal stresses as computed? (See Help 5.6.) [86]
5.6	Option 8 of menu 5.05 allows you to review all the results computed in this program segment. Included are the nodal displacements, element strains, element coordinate-direction stresses, element principal stresses, nodal coordinate stresses, nodal principal stresses, nodal reactions, and new nodal positions. Remember that you can use control-P whenever the screen is stopped to obtain a printed output of the results. [100]
5.7	You can list the nodal reactions as they are computed. [122]
5.8,5.9	As is the case for each of the program segments, you can specify the name for a data file. [152] Also, in conformity with the other programs you can specify the file you want to be retrieved. Free-run uses the default names. [152,162]
5.10,5.11	To aid you in recovery from disk errors, you can unlock a file or obtain a list of all the data files. [182,186]
5.12	This is the principal POSTPROCESS menu (5.05). Except when studying the computational details, elect option 1 which coordinates the execution of the program parts. Obtain the strains and stresses at the element nodes and centroids. As is true throughout the entire package, you can examine the results before proceeding (see Help 5.5). [212]
5.13	The exit menu routes you to PLOT, to the MAIN MENU of HELLO, back into POSTPROCESS, or out of the program entirely. [214]
5.14	If you did not elect the global free-run in the HELLO program, you are asked whether you wish to examine the progress of the calculational details. [226]

3.3.6 Plot

Help #	Comment [line #]
6.1	This general purpose subroutine allows you to decide when to proceed. Refer to the line cross-reference list for the origin of calls to this routine. [4]
6.2	When presented with graphical results (usually in the mixed graphics plus four lines of text), you can switch easily from Text (control-T), to mixed graphics (control-G), or to full view graphics (control-V) at any input statement in which the square brackets [] appear. The commands must appear as the first input character. (Note: the single keystroke input obtained through the Applesoft GET

statement does not support this control.) [90]

6.3 You can mark the highest and lowest values. (Note: the contour labelling option (Help 6.20) permits the actual contour values or letter codes to be given. [96]

6.4-6.7 The program presents the minimum and maximum data values for both contour and shading plots. The plot minimum and maximum falls within the range specified here. The default matches the extreme data values. You can select other values to produce more easily read contours or ranges. You can exclude one of the extreme values along a boundary to avoid an "extraneous" line on the plot resulting from the basic nature of the triangular element; for example, see the ideal fluid flow past a circular cylinder in a channel. A triangular element with linear interpolation over the element must necessarily have a constant value throughout if two of the three sides are assigned to the same constant boundary value. Consequently, every pair of two nodes on such a triangle technically falls on the contour. If you are interested in the computed value in the vicinity of such a corner triangle, use the diagonal reversal option in the mesh generation program GRID to force two triangles to share the corner vertex. The graphics resolution restricts you to eleven contour lines. (Note: any time your input is not accepted, you may use control-R to examine any restrictions.) [118,120]

6.8 If you are working with a complicated mesh and wish to increase the resolution of your hi-res graphics, subdivide the screen into equal-size rectangles and draw each of these subdivisions to fill the screen. Join screen dumps of the results for a more detailed view. Corner markers facilitate the re-combination process. [142]

6.9 In the upper-right corner of the screen is a map of the regions selected for enlargement. You can plot all zones, selected zones, or none. You select by designating those to be included or those to be excluded. This process overcomes the coarseness of the graphics at the expense of computational time. The entire figure is re-generated for each zone with the unneeded lines clipped (see Help 6.10 also). [144]

6.10 The zone plotting options provide great flexibility in getting increased resolution. If you want to enlarge a specific portion of the screen, select an arbitrary rectangular region to fill the entire screen. To avoid distortion this is subject to the maintenance of the width to height (aspect) ratio. Alternatively, you can generate multiple equal-sized zones (See Help 6.8 and 6.9). [176]

6.11 Redo the last zone plot if you are not satisfied. [180]

6.12 If you have made zone plots, you can re-select zones for further exploration. [184]

6.13 If you do not wish to save the plot or to take the time to label the plot, bypass the questions about labels for elements, nodes, contours, or captions. [192]

6.14,6.15 If you wish to add labels to elements or nodes, automatically label all or selectively label those of particular interest. To distinguish element and node numbers, an 'E' precedes element numbers (see Help 6.16 and 6.17). [194,196]

6.16,6.17 Use the left and right arrows to move the cursor from one label to the next. When pointing to one you wish to edit, press 'S' to select it. Use 'X' to add or remove the label (using the XDRAW command). To move the label in 'large' increments, use the traditional I,J,K,M diamond for up, left, right, or down movement. If you press the control key at the same time, control-I, control-J, control-K, and control-M move the cursor in smaller steps (1/10). When you have positioned the label correctly, press 'Q'. Select another label or press 'Q' a second time to quit editing. [204,206]

6.18 Since multiple lines and labels clutter a figure, you may choose to redraw the label. This makes the labels more prominent. Since labels are initially placed on the screen with the XDRAW command, those which overlap lines or other labels may become partly obscured. The redraw option first erases the screen where the label is currently placed then redraws the label. Unless over-written by another label, the legibility is now enhanced; however, you can no longer move the label. [206]

6.19-6.21 Identify a contour line by value or by a letter associated with a value in the legend (printed later on the text screen). If you wish to label a contour, use the left and right arrows to move the cursor from contour to contour. Press 'S' to select a contour to be labeled; then specify the actual location of the label using the I, J, K, and M diamond to move the label. Use 'X' to add or delete the label. 'Q' allows you to quit. [210,212]

6.22 When letters identify the contour lines, examine the legend on the text page by pressing control-T. Use control-V to return to the full graphics view. Then press return to continue. [214]

6.23 See Help 6.8 for a discussion of the redraw option. [216]

6.24-6.27 Now you can add text labels to the plot. See 6.19-6.21 for the commands. The only difference in the implementation is that you must provide the text of the label and initially locate your first guess for the center of the label. Note that you can elect to remove the label using 'X' and redraw later to enhance legibility. [218,220,222]

6.28 After generating a plot, you must specify whether to save the picture on the data diskette, obtain an enlargement, or simply abandon this picture.

Each picture requires 33 diskette sectors (of 256 bytes each) to save. If storing more than 8 to 10 figures (usually not necessary), you may need an additional data diskette. If you have a screen dump program resident in the $1D00.1FFF portion of memory, make a screen dump by simply pressing control-P. [226]

6.29 If you save the picture, the keyword will be a prefix to the filename selected and '.PIC' will be appended to identify this as a picture file. See Help 6.43 for a discussion of the use of the suffix. [228]

6.30 This requests the name of the data file to be loaded. The default is usually sufficient. The free-run mode uses the default. [258]

6.31,6.32 Use disk error recovery to unlock a file and to review the entire list of data file names. [278,282]

6.33 This is the principal PLOT menu (6.05) and provides an option for plotting each of the classes of calculated results. (Each of the earlier programs provided tabular listings; only plots are available in PLOT.)

(1) Plot the entire generated grid. See the earlier help comments about element, node, contour, and text labels, as well as zone plots.

(2) Plot the boundary outlines and indicate the boundary condition type. A boundary with no symbol added signifies a free surface. A triangle indicates a node constrained in both directions. 'Rollers' indicate that displacement is possible in only one direction. Arrows denote forces.

(3) Plot nodal displacements. Usually you plot the displaced mesh with solid lines and original mesh with dashed lines. If desired, use of these line types can be reversed. Alternatively, nodal displacements are indicated by vectors drawn from the original positions of the nodes. The various zoom and label options are available.

(4) and (5) Represent element stresses and strains by shading each element according to the magnitude of the element value. Principal stresses may be displayed as lines indicating the direction and relative magnitude of the selected component.

(6) Represent nodal stresses with contour plots or principal stress directions and relative magnitudes by lines.

(7) Finally, retrieve previously saved pictures for viewing or plotting if you have a screen dump capability. [316]

6.34 The exit menu has the usual options. This is frequently the program termination point. [318]

6.35 For nodal or element stresses, specify whether a coordinate-direction or principal stress is to be plotted . [326]

6.36 Select from among these coordinate-direction component options for plotting stresses and strains. [330]

6.37 Select from the maximum and minimum principal stresses and the maximum shear stress for plotting. [332]

6.38 Two plot type options show element stresses and strains.

(1) Shading shows the magnitude only.
(2) Lines show the magnitude and direction (for principal stresses only).
For nodal stress plots the options are:
(1) Lines to show direction and magnitude (for principal stresses only).
(2) Contour lines which show locations of equal stress values.
The contour line takes longer but can give higher resolution. Zoom plots can also be used to greatly enhance the resolution. [336]

6.39 Superimpose the plot on the boundary outline of the problem. Option 2 permits you to use a dashed mesh as background for the plot. This option can be effective as a diagnostic technique but often leads to figures which are 'too busy'. [338]

6.40-6.42 When plotting the displacement of nodes, you have several options. First decide whether to plot the entire displaced grid or just displacement vectors. If you decide to plot the displaced grid, elect to show either the entire displaced mesh superimposed on the original mesh or just the superimposed boundaries. Finally, decide which plot should be dashed and which solid. Usually you want the original plotted as dashed lines and the displaced plotted as solid lines. [342,344,346]

6.43,6.44 When you wish to retrieve a picture from the diskette, you have a list of file names which end with '.PIC'. Select one by entering only the menu item number, not the entire file name. This also displays the number of free sectors on the diskette. The picture is displayed in mixed text and graphics mode. Use control-T or control-V to change the display. Control-G returns you to the present display format.

Once you retrieve the results you can review them or print them if you have a 'Grappler' screen dump card. You can modify Line 320 to handle other screen dump programs or cards. [356,360]

3.3.7 Diskette Preparation

Help #	Comment [line #]
7.1	Remember to remove your program diskette. To avoid accidental erasure, always write protect (tab) the program diskette. [4]
7.2	Since you will be asked whether to erase the diskette, this is one last opportunity for you to review the names of the files. If this is not a formatted diskette, you will hear some unpleasant sounds; but the program will recover. [6]
7.3	You can abort the effort by deciding not to continue. In that case you are presented with the exit menu next. Otherwise, proceed with the diskette preparation.

When asked if you want to erase (re-format) the diskette, press 'Y' or 'N' and 'return'. 'Y' reformats the diskette as a data diskette. Neither a greeting program nor the Disk Operating System is on the diskette. Only sector 0 of track 0 and track 11 (i.e., decimal 17) are unavailable for data. Such a diskette does not boot-up the system. 'N' preserves the current contents of the diskette. If you remember to rename the 'FILEINFO.TXT' file first, you can place more than one problem on the diskette. Otherwise, the next step erases this vital information, rendering the existing problem unusable. [6]

7.4 To identify all of the related data files, use the keyword requested here. This short keyword precedes each data file name. The remainder of the file name simply indicates the nature of the data in the file. Choose a keyword which will help you identify the problem. [8]

7.5 The problem description allows you more flexibility in defining the problem. You can use up to 80 characters. You can use spaces freely; however, you cannot use a comma or a colon. By pressing 'return' you create the problem directory file called 'FILEINFO.TXT' erasing any previous file by that name. [8]

7.6,7.7 You can review the contents of the file created in this step. You can also review the contents of other random access and sequential text files in direct mode (i.e., when no program is executing) by issuing the command 'TDUMP filename'. Use control-S to stop and start the display. Use control-C to exit the file dump prematurely. [32]

7.8 The exit menu (7.10) permits you to return to the MAIN MENU in HELLO, to prepare another diskette, or to terminate the program. [34]

7.9 The program diskette must be in drive 1. Only the single drive users must swap diskettes here. [36]

7.10 This general subroutine awaits a user response to continue. [48]

3.4 Annotated Source Listings for Elasticity

3.4.0 Source Listings

HELLO.ELASTICITY

At boot-up, load the language card with DOS and part of the utilities.

```
2  TEXT :
   HOME :
   IF  PEEK (978) = 157 THEN
       PRINT "[M]LOADING THE LANGUAGE CARD[M][M][M]
          ONE MOMENT, PLEASE...[M][M][M]
          UTILITIES: PART 1":
       PRINT  CHR$ (4)"BLOAD FEM-B,A$2F70":
       CALL 12144:
       PRINT "[M][M]MOVING DOS TO BANK 2":
       PRINT  CHR$ (4)"BLOAD FEM-A,A$4000":
       CALL 16384:
       REM HELLO(E)
```

Place the 'Hello' program above hi-res page 1.

```
4  VTAB 12:
   PRINT "LOCATE APPLESOFT ABOVE HI-RES PAGE 1":
   PRINT  CHR$ (4)"BLOAD LOADER":
   CALL 768:
   D$ =  CHR$ (4):
   WR$ = "... WHEN READY, PRESS <RETURN>":
   SR$ = "... SELECT BY NUMBER <RETURN>":
   ON  PEEK (2051) = 169 GOTO 8
```

If not already loaded, put additional utilities below hi-res page 1.

```
6  PRINT "[M][M]LOADING UTILITIES (PART 2) AND SYMBOLS":
   PRINT D$"BLOAD FEM-C,A$803,D1":
   CALL 2051:
   & D(T):
   POKE 2165,1
```

Identify the program and, if necessary, reload the shape table and hi-res dump.

```
8  ON  PEEK (6462) = 75 AND  PEEK (6463) = 75 GOTO 10:
   PRINT D$"BLOAD SHAPES,A$193E,D1":
   PRINT D$"BLOAD HI-RES DUMP,A$1D00,D1"
```

If using a II-e or II-c, remind to set 'caps lock'.

```
10 POKE 2164,0:
   HOME :
   POKE 232,62:
   POKE 233,25:
   IF  PEEK (64435) = 6 THEN
       PRINT "[M][M]SET CAPS LOCK AND":
       & I(WR$;"";A$,"1")
```

Set the number of disk drives. If a dual drive, insert the data diskette.

```
12 A =  PEEK (2048):
   ON A = 1 OR A = 2 GOTO 14:
   HOME :
   & B(20,20):
   & I("[M][M]HOW MANY DISK DRIVES ARE YOU USING";
     2;A,"2",0,0,A = 1 OR A = 2):
   POKE 2048,A:
   ON A = 1 GOTO 14:
```

```
   & B(200,20):
   PRINT "[M][M]INSERT ";:
   FLASH :
   PRINT "DATA";:
   NORMAL :
   PRINT " DISK INTO DRIVE 2[M][M]":
   & I(WR$;"";A$,"3")
```

(14-16) Set the level of user interaction.

```
14 HOME :
   PRINT "* USER INTERACTION MODE *[M][M]
      1. MAXIMAL (NO DEFAULTS)[M]
      2. INTERMEDIATE (SOME DEFAULTS)[M]
      3. MINIMAL (FREE RUN)[M]
      4. DEMO WITH PREVIOUS DATA[M]":
   & I(SR$,2;A,"4",BC%,1,A >0 AND A <5):
   ON BC% GOTO 12:
   POKE 2163,A - 1

16 IF A = 4 THEN
      & I("[M][G]WARNING:
      CALCULATED DATA WILL NOT BE[M] SAVED! OK? (Y/N)";
      "N";A$,"5",0,1): ON A$ < >"Y" GOTO 14
```

Main title.

```
18 HOME :
   INVERSE :
   FOR I = 1 TO 4:
      PRINT  TAB( 2)" " TAB( 38)" ";
   NEXT I:
   NORMAL :
   VTAB 2:
   HTAB 3:
   PRINT " APPLIED FINITE ELEMENT ANALYSIS: ":
   HTAB 3:
   PRINT "     AN APPLE " CHR$ (221) CHR$ (219);
     " IMPLEMENTATION     ":
   VTAB 6

20 HTAB 15:
   INVERSE :
   PRINT "ELASTICITY[M][M]":
   NORMAL :
   PRINT  TAB( 19)"BY[M][M]" TAB( 7)
     "J. R. COOKE AND D. C. DAVIS[M][M][M][M]"
     TAB( 13)"COPYRIGHT, 1985[M][M]" TAB( 8)
     "JOHN WILEY AND SONS, INC.[M][M]" TAB( 11)
     "ALL RIGHTS RESERVED":
   VTAB 23:
   & I(WR$;"";A$,"6",BC%):
   ON BC% < >0 GOTO 10

22 DATA  "GEOMETRY","DEFINE PROBLEM GEOMETRY",3,"GRID",
     "GENERATE & MODIFY GRID",5,"PREPROCESS",
     "DEFINE PROPERTIES AND BOUNDARY       CONDITIONS",7,
     "SOLVE","ASSEMBLE EQUATIONS & SOLVE FOR NODE DISPLACEMENTS",
     10,"POSTPROCESS"

24 DATA "SOLVE FOR ELEMENT STRAINS & STRESSES
     & NODE STRESSES",13,"PLOT",
     "PLOT GRID, BOUNDARY CONDITIONS,
     &    PROBLEM RESULTS",16,"DISKETTE PREPARATION",
     "PREPARE DATA DISKETTE",19:
```

```
        RESTORE :
        FOR I = 1 TO 7:
            READ PN$(I),PD$(I),LI(I):
        NEXT I
```

The MAIN menu.

```
26  N = PEEK (2165):
    HOME :
    PRINT "***    MAIN PROGRAM MENU (0.00)      ***[M]":
    FOR I = 1 TO 7:
        PRINT I". ";:
        INVERSE :
        PRINT PN$(I):
        NORMAL :
        PRINT "    " LEFT$ (PD$(I),36);:
        IF LEN (PD$(I)) >36 THEN
            PRINT "    " RIGHT$ (PD$(I), LEN (PD$(I)) - 36);

28      PRINT :
    NEXT I:
    PRINT "[M]0. STOP[M]":
    POKE 34,22:
    & I(SR$;N;N,"7",0,1,N > = 0 AND N <8):
    POKE 34,0:
    IF N = 0 THEN
        END

30  IF PEEK (2163) = 2 THEN
        & B(10,10):
        & B(0,0,10)
```

(32-36)Activate the selected program.

```
32  VTAB LI(N):
    INVERSE :
    PRINT N". "PN$(N) TAB( 40)" ":
    POKE 2165,N:
    POKE 2166,1:
    NORMAL :
    IF N = 4 OR N = 5 THEN
        POKE 103,63:
        POKE 104,25:
        POKE 6462,0:
        REM    $193E

34  ONERR GOTO 38

36  PRINT D$:
    PRINT D$"RUN "PN$(N);DR$(DR)
```

Error handling.

```
38  POKE 34,22:
    VTAB 23:
    POKE 216,0:
    HOME :
    PRINT CHR$ (7) CHR$ (7):
    & I("INSERT CORRECT PROGRAM DISK <RETURN>";
      "";A$,"8"):
    GOTO 36

65535 REM [M][M]1FEB85[M]JRC/DCD

END-OF-LISTING
```

GEOMETRY.ELASTICITY

See the identical GEOMETRY.HEAT source listing in Chapter 2, page 46.

GRID.ELASTICITY

See the identical GRID.HEAT source listing in Chapter 2, page 74.

PREPROCESS.ELASTICITY

(1,228-254) Cold start initialization.

```
1  TEXT :
   & L(255):
   P% = 10:
   S$ = "STARTING":
   E$ = "ENDING":
   P$ = " NODE ":
   U$ = "( LEFT ARROW  RIGHT ARROW S Q )  ND/EL  PRESS
      <S>AT ":
   SR$ = "... SELECT BY NUMBER <RETURN>":
   WR$ = "... WHEN READY, PRESS <RETURN>":
   FI$ = "FILEINFO.TXT":
   D$ =  CHR$ (4):
   GOTO 228:
   REM PRE(E)
```

(2,256-258) Warm restart.

```
2  GOTO 256
```

Wait for user response, clear screen, and continue.

```
4  PRINT :
   & B(20,20):
   VTAB 24:
   & I(WR$;"";A$,"1",BC%):
   HOME :
   RETURN
```

Free-run pause.

```
6  ON Z < >2 GOTO 4:
   & B(10,10):
   & B(0,0,10):
   HOME :
   RETURN
```

Find distance between two points and the unit normal vector.

```
8  X1 = XN(N1%):
   Y1 = YN(N1%):
   X2 = XN(N2%):
   Y2 = YN(N2%):
   L =  SQR ((X1 - X2) ^ 2 +  (Y1 - Y2) ^ 2):
   NX = (Y2 - Y1) / L:
   NY = (X1 - X2) / L:
   RETURN
```

Draw symbol at a node and get instructions.

```
10  SCALE= 1:
    NO = BN%(N,1):
    VTAB 23:
    HTAB 37:
    PRINT "   ";:
    HTAB 37:
    PRINT NO:
    XDRAW 67 AT  FN PX(XN(NO)), FN PY(YN(NO)):
    GET A$:
    A =  ASC (A$):
    XDRAW 67 AT  FN PX(XN(NO)), FN PY(YN(NO)):
    N = N +  FN M(B):
```

```
    ON A$ < >"S" AND A$ < >"Q" GOTO 10:
    RETURN
```

Find node for boundary condition deletion.

```
12  ON NI = 0 GOTO 18:
    POKE 34,21:
    HOME :
    FOR NA = 1 TO NI:
        NO = BN%(N,1):
        A0 = (BI%(NA,1) = N AND BI%(NA,3)
                          = BD AND BI%(NA,2) = BT):
        ON A0 GOSUB 14:
    NEXT NA:
    HOME :
    RETURN
```

Delete a boundary condition at a node and update array.

```
14  GOSUB 46:
    FOR I = NA TO NI - 1:
        FOR J = 1 TO 3:
            BI%(I,J) = BI%(I + 1,J):
        NEXT J:
        BV(I) = BV(I + 1):
    NEXT I:
    FOR J = 1 TO 3:
        BI%(NI,J) = 0:
    NEXT J:
    BV(NI) = 0:
    NI = NI - 1:
    RETURN
```

Add a boundary condition and plot its symbol.

```
16  NI = NI + 1:
    BI%(NI,1) = N:
    BI%(NI,2) = BT:
    BI%(NI,3) = BD:
    BV(NI) = VB:
    NA = NI:
    NO = BN%(N,1):
    GOSUB 46

18  RETURN
```

Select a node for addition or deletion of a single boundary condition.

```
20  HOME :
    N = FN N(N):
    PRINT U$" DESIRED POINT":
    GOSUB 10:
    ON A$ = "Q" GOTO 18:
    ON AC GOSUB 16,12:
    RETURN
```

(22-26) Select a range of boundary nodes for addition or removal of a force or displacement boundary condition.

```
22  HOME :
    N = FN N(N):
    PRINT U$;:
    INVERSE :
    PRINT S$;:
    NORMAL :
    PRINT P$:
    GOSUB 10:
    ON A$ = "Q" GOTO 18:
```

```
    N1 = N:
    ON AC GOSUB 16,12

24  HOME :
    PRINT U$;:
    INVERSE :
    PRINT E$;:
    NORMAL :
    PRINT P$:
    GOSUB 10:
    ON A$ = "Q" OR N = N1 GOTO 18:
    N2 = N:
    ON AC GOSUB 16,12:
    A = 21:
    N = N1:
    N = N1 +  FN M(B):
    ON N = N2 GOTO 18

26  ON AC GOSUB 16,12:
    N = N +  FN M(B):
    ON N < >N2 GOTO 26:
    RETURN
```

(28-32) Select a range of the boundary for addition or removal of a surface stress boundary condition.

```
28  HOME :
    N =  FN N(N):
    PRINT U$;:
    INVERSE :
    PRINT S$;:
    NORMAL :
    PRINT P$:
    GOSUB 10:
    ON A = 81 GOTO 18:
    N1 = N

30  HOME :
    PRINT U$;:
    INVERSE :
    PRINT E$;:
    NORMAL :
    PRINT P$:
    GOSUB 10:
    ON A = 81 GOTO 18:
    N2 = N:
    ON N1 = N2 GOTO 18:
    N = N1:
    A = 21

32  ON AC GOSUB 16,12:
    N = N +  FN M(B):
    ON N < >N2 GOTO 32:
    RETURN
```

(34-36) Convert coordinate direction surface stress to a nodal force boundary condition.

```
34  N = BI%(NA,1):
    EL = BN%(N,2):
    DI = BI%(NA,3):
    A = 21:
    N0 = N +  FN M(B):
    N1% = BN%(N,1):
    N2% = BN%(N0,1):
    GOSUB 8:
    TH = TH(EL) * (TE = 1) +  P2 * (TE = 2) / 6
```

```
36  F = L * TH * BV(NA):
    W1 = .5 * (TE = 1) +  (2 * X1 +  X2) * (TE = 2):
    W2 = .5 * (TE = 1) +  (X1 +  2 * X2) * (TE = 2):
    IF DI <3 THEN
        D1 = 2 * N1% - 2 +  DI:
        BC(D1,2) = BC(D1,2) +  W1 * F:
        D2 = 2 * N2% - 2 +  DI:
        BC(D2,2) = BC(D2,2) +  W2 * F:
        BC(D1,1) = 1:
        BC(D2,1) = 1:
        RETURN
```

Convert a normal or a tangent surface stress to a nodal force boundary condition.

```
38  NU = NX * (DI = 3) - NY * (DI = 4):
    NV = NY * (DI = 3) +  NX * (DI = 4):
    D2 = 2 * N1%:
    D1 = D2 - 1:
    BC(D1,2) = BC(D1,2) +  W1 * NU * F:
    BC(D2,2) = BC(D2,2) +  W1 * NV * F:
    BC(D1,1) = 1:
    BC(D2,1) = 1:
    D2 = 2 * N2%:
    D1 = D2 - 1:
    BC(D1,2) = BC(D1,2) +  W2 * NU * F:
    BC(D2,2) = BC(D2,2) +  W2 * NV * F:
    BC(D1,1) = 1:
    BC(D2,1) = 1:
    RETURN
```

Add a point force input boundary condition to the equivalent boundary condition array.

```
40  N = BI%(NA,1):
    DI = BI%(NA,3):
    N1 = BN%(N,1):
    DF = 2 * N1 - 2 +  DI:
    BC(DF,2) = BC(DF,2) +  BV(NA):
    BC(DF,1) = 1:
    RETURN
```

Assign fixed displacement constraint to equivalent boundary condition array.

```
42  N = BI%(NA,1):
    DI = BI%(NA,3):
    N1 = BN%(N,1):
    DF = 2 * N1 - 2 + DI:
    BC(DF,1) = 2:
    BC(DF,2) = BV(NA):
    RETURN
```

Optional display of data filenames.

```
44  & I("[M]CATALOG?";"N";A$,"2",0,1):
    ON A$ = "N" GOTO 18:
    PRINT D$"CATALOG"DR$(DR):
    RETURN
```

(46-50) Draw symbol for a point force or displacement boundary condition at a node.

```
46  VB = BV(NA):
    N = BI%(NA,1):
    NO = BN%(N,1):
    BT = BI%(NA,2):
    DI = BI%(NA,3):
    ON BT = 3 GOTO 52:
    A = 8:
    B1 = N +   FN M(B):
    A = 21:
    B2 = N +   FN M(B):
```

```
    N1% = BN%(B1,1):
    N2% = BN%(B2,1):
    GOSUB 8:
    IF BT = 0 OR (BT = 1 AND VB = 0) THEN
         ROT= 8 * (DI = 2):
        SCALE= 2 +  SZ:
        SY = 64:
        GOTO 50

48  A0 = NX * (DI = 1) +  NY * (DI = 2):
    ROT= 48 * (DI = 2) +  64 * (DI = 1) - 32
       * ((BT = 1 AND VB <0) OR (BT = 2 AND A0 <0)):
    SY = 73 * (BT = 2) +  71 * (BT = 1)
       - (BT = 1 AND VB * A0 >0):
    SCALE= 1

50  XDRAW SY AT  FN PX(XN(NO)) +  (BT < >0)
       * NX, FN PY(YN(NO)) - (BT < >0) * NY:
    SCALE= 1:
    ROT= 0:
    RETURN
```

(52-54) Draw symbol for a surface stress boundary condition.

```
52  A = 21:
    N1% = NO:
    N2% = BN%(N +   FN M(B),1):
    GOSUB 8:
    NU = (DI = 1) +  NX * (DI = 3) - NY * (DI = 4):
    NV = (DI = 2) +  NY * (DI = 3) +  NX * (DI = 4):
    X1 =  FN PX((X1 +  X2) / 2):
    Y1 =  FN PY((Y1 +  Y2) / 2)

54  ROT= 0:
    FOR I = 1 TO 3:
        A0 = (I = 1 AND VB <0) +  (I = 3 AND VB > = 0):
        SY = 74 - 10 * A0:
        SCALE= 1 +  A0:
        XDRAW SY AT X1 +  2 * I * NU,Y1 - 3 * I * NV:
    NEXT I:
    SCALE= 1:
    RETURN
```

(56-58) Plot boundaries of the problem domain and label the axis directions.

```
56  B0 = B:
    B1 = I1:
    B2 = I2:
    HGR :
    & D(G):
    HCOLOR= 3:
    A = 21:
    FOR B = 1 TO NS:
        I1 = IS(2 * B - 1):
        I2 = IS(2 * B):
        FOR N = I1 TO I2:
            N0 = N +   FN M(B):
            N1% = BN%(N,1):
            N2% = BN%(N0,1):
            HPLOT  FN PX(XN(N1%)), FN PY(YN(N1%))
               TO  FN PX(XN(N2%)), FN PY(YN(N2%)):
    NEXT N,B:
    B = B0:
    I1 = B1:
    I2 = B2
```

```
58  SCALE= 1:
    ROT= 48:
    DRAW 70 AT 5,90:
    ROT= 0:
    DRAW 70 AT 130,155:
    DRAW  ASC (DI$(1)) - 31 AT 148,158:
    DRAW  ASC (DI$(2)) - 31 AT 2,75:
    RETURN
```

Draw an element and label or fill to indicate the region.

```
60  FOR I = 1 TO DE:
        N1 = ND%(EL,I):
        N2 = ND%(EL, FN P2(I)):
        X1 =  FN PX(XN(N1)):
        Y1 =  FN PY(YN(N1)):
        X2 =  FN PX(XN(N2)):
        Y2 =  FN PY(YN(N2)):
        HPLOT X1,Y1 TO X2,Y2:
    NEXT I:
    L$ =  STR$ (EL):
    XP =  FN CX(EL):
    YP =  FN CY(EL):
    ON IO = 2 GOTO 66:
    & F(CO%(RE),OP%(RE),XP,YP):
    RETURN
```

Determine the boundary B on which node N occurs.

```
62  B = B +  1:
    I1 = IS(2 * B - 1):
    I2 = IS(2 * B):
    ON N > = I1 AND N < = I2 GOTO 18:
    GOTO 62
```

Draw all boundary conditions.

```
64  SZ = 0:
    FOR NA = 1 TO NI:
        N = BI%(NA,1):
        B = 0:
        GOSUB 62:
        GOSUB 46:
    NEXT NA:
    SZ = 1:
    RETURN
```

Draw label L$ on graphics screen.

```
66  FOR J1 = 1 TO  LEN (L$):
        XDRAW  ASC ( MID$ (L$,J1,1)) - 31 AT XP - 7
           *  LEN (L$) / 2 +  7 * (J1 - 1),YP +  3.5:
    NEXT J1:
    RETURN
```

Erase background for label L$.

```
68  HCOLOR= 0:
    FOR J1 = 1 TO 9:
        HPLOT XP - 3.5 *  LEN (L$) - 1,YP - 4.5
           +  J1 TO XP +  3.5 *  LEN (L$),YP - 4.5 +  J1:
    NEXT J1:
    HCOLOR= 3:
    GOTO 66
```

(70-110) Input reference temperature for thermal expansion calculations, specify uniformity of property, and define for entire domain if uniform.

```
70  HOME :
    FOR I = 1 TO NE:
```

```
            MP(I,0) = I:
        NEXT I:
        T1 = 1:
        T2 = 5:
        ON ED GOTO 112

72      FOR TY = T1 TO T2:
            ON TY = 3 AND TE = 2 GOTO 110:
            E1 = 1:
            E2 = NE:
            R1 = 1:
            R2 = NQ:
            POKE 34,2:
            HOME :
            PRINT  TAB( (40 -  LEN (TY$(TY)) - 10) / 2);
              "PROPERTY: ";:
            INVERSE :
            PRINT TY$(TY):
            NORMAL :
            POKE 34,3


74          IF TY = 5 THEN
                & I("[M][M]THERMAL EXPANSION REFERENCE TEMPERATURE:";
                  MP(0,0);MP(0,0),"3",BC%):
                ON BC% GOTO 74,108,74

76          PRINT "[M]"TY$(TY)" IS UNIFORM FOR[M]
              1. ENTIRE BODY[M]2. "IO$(1)"[M]3. "IO$(2)"[M]":
            & I(SR$,1;IO,"4",BC%,0,IO >0 AND IO < 4)

78          TY = TY * (BC% < >1) +  (BC% = 1)
              * (TY - 2) * (TY >T1):
            IF BC% = 1 THEN
                NEXT TY

80          ON BC% GOTO 70,108,76:
            UN = (IO = 1):
            ON UN < >1 GOTO 84:
            HOME :
            INVERSE :
            PRINT "[M]"TY$(TY)" IS UNIFORM[M]IN THE BODY.":
            NORMAL :
            PRINT "[M]WHAT IS THE "TY$(TY):
            IF TY < >2 THEN
                & I("VALUE";MP(1,TY);MP(1,TY),"5",BC%)

82          IF TY = 2 THEN
                & I("VALUE (0 <= P <.5)";
                  MP(1,TY);MP(1,TY),"6",BC%,0,MP(1,TY) > = 0
                      AND MP(1,TY) <.5)

84          ON BC% GOTO 76,108,76:
            E0 = 1:
            R0 = 1:
            MP(0,TY) = UN:
            IF UN = 1 THEN
                FOR EL = 1 TO NE:
                    MP(EL,TY) = MP(1,TY):
            NEXT EL:
            GOTO 104
```

(86-94) Input coded values for property type.

```
86          HOME :
```

```
         IO = IO - 1:
         INVERSE :
         PRINT "[M]"TY$(TY)"[M]UNIFORM IN "IO$(IO)"-":
         NORMAL :
         & I("[M]HOW MANY CODED " + TY$(TY)
            + "[M]VALUES? (0-6)";2 * (IO < >1
            OR (IO = 1 AND R2 >6)) + R2 * (IO = 1
            AND R2 <7);NC,"7",BC%,0,NC > = 0
            AND NC <7)

88       ON BC% GOTO 76,108,86:
         ON NC = 0 GOTO 96:
         PRINT "[M]ASSIGN CODE VALUES":
         INVERSE :
         PRINT " CODE     "TY$(TY);:
         IF TY = 2 THEN
            PRINT " ( 0 <= P <.5)";

90       NORMAL :
         PRINT :
         FOR I = 1 TO NC:
             PRINT "  " CHR$ (64 + I) TAB( 10);:
             IF TY = 2 THEN
                  & I("";CV(I,TY);CV(I,TY),"8",BC%,0,
                     CV(I,TY) > = 0 AND CV(I,TY) <.5)

92           IF TY < >2 THEN
                  & I("";CV(I,TY);CV(I,TY),"9",BC%)

94           I = I + NC * (BC% = 2) - (1 + (I >1))
                * (BC% = 1) - (BC% = 3):
         NEXT I:
         ON BC% = 2 GOTO 108
```

96-102) Specify property values by element.

```
96       GOSUB 56:
         POKE 34,20:
         HOME :
         ON IO = 1 GOTO 120:
         ON NC >0 GOSUB 118:
         FOR EL = E1 TO E2:
             EO = EL:
             GOSUB 60:
             MP$ = "A":
             ON NC >0 GOTO 98:
             MP$ = STR$ (MP(EL,TY))

98           POKE 34,22:
             HOME :
             PRINT TY$(TY)" FOR ELEMENT "EL:
             & I("CODE / # / Q ";MP$;A$,"10",BC%):
             HOME :
             ON BC% GOTO 98,102,98:
             ON LEN (A$) = 0 GOTO 102:
             A = ASC (A$):
             BC% = 2 * (A = 81):
             ON (A >47 AND A <58) OR A = 46
                OR (A >64 AND A <65 + NC)
                OR A = 81 GOTO 100:
             & B(200,20):
             GOTO 98

100          EL = (EL - 1) * (BC% = 1) + NE * (BC% = 2)
                + EL * (BC% = 0):
```

```
            ON BC% GOTO 98,102,98:
            MP(EL,TY) =  VAL (A$) * (A >45 AND A <58)
               +  CV((A - 64) * (A >64 AND A <65
               +  NC),TY) * (A >64 AND A <65 +  NC)

102         A0 = (BC% < >2 AND A < >81):
            ON A0 GOSUB 60:
            L$ =  CHR$ (A * (A >64) +  42 * (A <65)):
            ON A0 GOSUB 66:
            EL = EL +  E2 * (A0 = 0):
        NEXT EL:
        GOSUB 4:
        & D(T):
        POKE 34,2:
        HOME
```

(104-108) Specify edit scope for property type.

```
104     & I("[M]EDIT " +  TY$(TY)
           +  " VALUES";"N";A$,"11",0,1):
        ON A$ = "N" GOTO 108:
        PRINT "[M]EDIT SCOPE: UNIFORM BY[M][M]
           1. "IO$(1)"[M]2. "IO$(2)"[M]":
        & I(SR$,IO;IO,"12",BC%,0,IO >0 AND IO <3):
        ON BC% GOTO 104,108,104

106     EL = E0 * (E0 < = NE) +  (E0 >NE):
        RE = R0 * (R0 < = NQ) +  (R0 >NQ):
        A = EL * (IO = 2) +  RE * (IO = 1):
        MX = NE * (IO = 2) +  NQ * (IO = 1):
        & I("[M]START AT WHAT " +  IO$(IO);A;A,"13",
           BC%,0,A > = 0 AND A < = MX):
        RE = A:
        EL = A:
        ON A = 0 GOTO 74:
        E1 = EL:
        R1 = RE:
        IO = IO +  1:
        POKE 34,2:
        GOTO 86
```

Check for uniformity of property.

```
108     POKE 34,2:
        HOME :
        PRINT "UNIFORMITY CHECK OF[M]"TY$(TY)"...":
        & O(MP,1 TO NE,TY,A):
        MP(0,TY) = (MP(1,TY) = MP(NE,TY)):
        & O(MP,1 TO NE,0,A):
        HOME :
        PRINT "[M][M]"TY$(TY)" IS "UN$(MP(0,TY))"UNIFORM":
        GOSUB 6:
        ON BC% GOTO 104:
        TY = TY +  5 * (BC% = 2)

110 NEXT TY
```

(112-116) Select property edit.

```
112 ON Z = 2 GOTO 18:
    HOME :
    T1 = (TY - 1) * (TY >0) +  (TY = 0):
    PRINT "[M][M]EDIT PROPERTIES:[M]":
    FOR I = 1 TO 5:
       PRINT I". "TY$(I):
    NEXT I:
    PRINT "[M]0. NONE[M]":
```

```
    & I(SR$,0;T1,"14",BC%,0,(T1 >= 0
      AND T1 <6) AND  NOT (TY = 3 AND TE = 2))

114 ON T1 = 0 GOTO 116:
    T2 = T1 * (ED = 1) +  5 * (ED = 0):
    GOTO 72

116 POKE 34,0:
    POKE 2166,2:
    PRINT "[M][M]END OF MATERIAL PROPERTIES":
    GOSUB 6:
    ON BC% >0 GOTO 112:
    RETURN
```

Display coded property values.

```
118 POKE 34,20:
    HOME :
    FOR I = 1 TO NC:
        VTAB 21 +  (I >3):
        HTAB ((I - 3) * (I >3) +  I * (I <4)) * 12 - 10:
        PRINT  CHR$ (64 +  I)"="CV(I,TY):
    NEXT I:
    RETURN
```

(120-126) Specify property values by input region.

```
120 ON NC >0 GOSUB 118:
    FOR RE = R1 TO R2:
        R0 = RE:
        FOR EL = 1 TO NE:
            ON ND%(EL,0) = RE GOSUB 60:
        NEXT EL:
        L$ =  STR$ (RE):
        XP = CR(RE,1):
        YP = CR(RE,2):
        GOSUB 68:
        RP$ = "A":
        ON NC >0 GOTO 122:
        RP$ =  STR$ (RP(RE,TY))

122     POKE 34,22:
        HOME :
        PRINT TY$(TY)" FOR REGION "RE:
        & I("CODE / # / Q ";RP$;A$,"15",BC%):
        HOME :
        ON BC% GOTO 122,126,122:
        ON  LEN (A$) = 0 GOTO 126:
        A =  ASC (A$):
        BC% = 2 * (A = 81):
        ON (A >47 AND A <58) OR A = 46
           OR (A >64 AND A <65 +  NC)
           OR A = 81 GOTO 124:
        & B(200,20):
        GOTO 122

124     RE = (RE - 1) * (A = 60) +  NQ * (A = 81) + RE
           * (BC% <1 OR BC% >2):
        ON BC% GOTO 122,126,122:
        IF  LEN (A$) >0 THEN
            RP(RE,TY) =  VAL (A$) * (A <65
               OR A >64 +  NC) +  CV((A - 64)
               * (A >64 AND A < = 64 +  NC),TY)
               * (A >64 AND A < = 64 +  NC)
```

```
126       L$ =  STR$ (RE):
          A0 = (BC% < >2 AND A < >81):
          ON A0 GOSUB 66:
          L$ =  CHR$ (A * (A >64) +  42 * (A <65)):
          ON A0 GOSUB 66:
          RE = RE +  R2 * (A0 = 0):
     NEXT RE:
     FOR I = 1 TO NE:
          MP(I,TY) = RP(ND%(I,0),TY):
     NEXT I:
     GOSUB 4:
     & D(T):
     POKE 34,2:
     HOME :
     GOTO 104
```

(128-132) Initialize, give instructions for boundary conditions, and plot boundary.

```
128  HOME :
     SZ = 1:
     ON ED GOTO 130:
     NI = 0

130  POKE 34,2:
     & D(T):
     PRINT "INSTRUCTIONS (3.21):[M]
        BOUNDARY CONDITIONS MAY BE DEFINED OR  DELETED
        AT A NODE OR RANGE OF NODES FOR EACH DIRECTION
        COMPONENT. COUNTERCLOCK- WISE ORDER MUST BE
        USED FOR RANGES.[M][M]"

132  GOSUB 6:
     ON BC% GOTO 18,18,130:
     GOSUB 56:
     B = 1:
     I1 = IS(2 * B - 1):
     I2 = IS(2 * B):
     ON ED GOSUB 64
```

(134-154) Specify boundary condition type, direction, value, and segment of boundary where applied.

```
134  POKE 34,20:
     HOME :
     PRINT "DEFINE BOUNDARY CONDITIONS-[M]
        OPTIONS: ENTER, DELETE, QUIT, BRANCH":
     & I("E D Q < >";"E";A$,"16",BC%,1,
        A$ = "E" OR A$ = "D" OR A$ = "Q"):
     ON BC% GOTO 130,154,134:
     ON A$ = "Q" GOTO 154:
     AC = (A$ = "E") +  2 * (A$ = "D")

136  HOME :
     PRINT AC$(AC)"-":
     POKE 3,21

138  HOME :
     PRINT "1. "BT$(1) TAB( 20)"3. "BT$(3)"[M]2. "BT$(2):
     & I("1 2 3 Q <>";"1";A$,"17",BC%,1,
        A$ = "1" OR A$ = "2" OR A$ = "3" OR A$ = "Q"):
     ON BC% GOTO 134,134,138:
     BT =  VAL (A$):
     ON A$ = "Q" GOTO 134

140  POKE 34,20:
```

```
      HOME :
      PRINT AC$(AC)" "BT$(BT)"-":
      POKE 34,21:
      ON BT GOTO 146,146,142

142   HOME :
      PRINT "1. "BD$(1) TAB( 20)"3. "BD$(3)"[M]
        2. "BD$(2) TAB( 20)"4. "BD$(4):
      & I("1 2 3 4 Q <>";"1";A$,"18",BC%,1,
        A$ = "1" OR A$ = "2" OR A$ = "3" OR
         A$ = "4" OR A$ = "Q"):
       ON BC% GOTO 138,134,142:
       ON A$ = "Q" GOTO 134

144   BD = VAL (A$):
      POKE 34,20:
      HOME :
      PRINT AC$(AC)" "BD$(BD)" "BT$(BT)"-":
      VTAB 22:
      GOTO 148

146   HOME :
      PRINT "1. "BD$(1) TAB( 20)"2. "BD$(2):
      & I("1 2 Q <>";"1";A$,"19",BC%,1,A$
         = "1" OR A$ = "2" OR A$ = "Q"):
      ON BC% GOTO 138,134,146:
      ON A$ = "Q" GOTO 134:
      BD = VAL (A$):
      POKE 34,20:
      HOME :
      PRINT AC$(AC)" "BD$(BD)" "BT$(BT)"-":
      VTAB 22

148   POKE 34,21:
      ON AC = 2 GOTO 150:
      HOME :
      & I(" " + BT$(BT) + " VALUE = ";0;VB,"20",BC%):
      ON BC% GOTO 146,134,148:
      POKE 34,20:
      HOME :
      PRINT BD$(BD)" "BT$(BT)" = "VB:
      POKE 34,21

150   IF BT = 3 THEN
          GOSUB 28:
        GOTO 134

152   HOME :
      PRINT "SPECIFY WHERE TO " + AC$(AC) + " B.C.[M]
         1. SINGLE POINT, OR 2. CONTINUOUS":
      & I("1 2 Q <>";"1";A$,"21",BC%,1,
         A$ = "1" OR A$ = "2" OR A$ = "Q"):
      ON BC% GOTO 148,134,152:
      ON A$ = "Q" GOTO 134:
      A0 = VAL (A$):
      ON A0 GOSUB 20,22:
      GOTO 134

154   IF NS >1 THEN
          POKE 34,20:
        HOME :
        & I("ANOTHER BOUNDARY";"Y";A$,"22",BC%,1):
        HOME :
```

```
    IF A$ = "Y" THEN
        PRINT " BOUNDARY #";:
        GET A$:
        B =  VAL (A$):
        ON B <1 OR B >NS GOTO 154:
        I1 = IS(2 * B - 1):
        I2 = IS(2 * B):
        GOTO 134
```

Edit boundary conditions.

```
156 TEXT :
    & D(T):
    HOME :
    ON Z = 2 GOTO 158:
    PRINT "EDIT BOUNDARY CONDITIONS[M][M]":
    & I("CHANGES? (Y/N)";"N";A$,"23",0,1):
    ED = (A$ = "Y"):
    ON ED GOTO 128
```

(158-162) Save input boundary conditions (RE=10).

```
158 ON NI = 0 GOTO 18:
    & C(BI):
    DIM BI(NI,4):
    FOR I = 1 TO NI:
        FOR J = 1 TO 3:
            BI(I,J) = BI%(I,J):
        NEXT J:
        BI(I,4) = BV(I):
    NEXT I:
    RE = 10:
    GOSUB 202:
    GOSUB 196:
    N1 = NI:
    N2 = 4:
    ONERR  GOTO 218

160 EF = 0:
    GOSUB 198:
    ON  PEEK (2163) = 3 GOTO 162:
    & S(BI,N$)

162 ON EF GOTO 160:
    GOSUB 200:
    & C(BI):
    GOSUB 6:
    PRINT "ASSEMBLING BOUNDARY CONDITIONS FOR EACH
      DEGREE OF FREEDOM";
```

(164-170) Save nodal equivalent boundary conditions (RE=11).

```
164 & C(BC):
    DIM BC(NP,2):
    FOR NA = 1 TO NI:
        PRINT ".";:
        ON BI%(NA,2) = 3 GOSUB 34:
    NEXT NA:
    FOR NA = 1 TO NI:
        PRINT ".";:
        ON BI%(NA,2) = 1 GOSUB 40:
    NEXT NA:
    FOR NA = 1 TO NI:
        PRINT ".";:
        ON BI%(NA,2) = 2 GOSUB 42:
    NEXT NA
```

```
166  PRINT :
     RE = 11:
     GOSUB 196:
     N1 = NP:
     N2 = 2:
     ONERR  GOTO 218

168  EF = 0:
     GOSUB 198:
     ON  PEEK (2163) = 3 GOTO 170:
     & S(BC,N$):
     ON EF GOTO 168

170  GOSUB 200:
     & C(BC,TH):
     R1 = 12:
     GOTO 216
```

(172-174) Find region centroids.

```
172  NS = BN%(0,0):
     & C(CR):
     DIM CR(NQ,2):
     PRINT "[M]FINDING REGION CENTROIDS";:
     FOR EL = 1 TO NE:
          RE = ND%(EL,0):
          PRINT ".";:
          FOR N = 1 TO DE:
               N1 = ND%(EL,N):
               CR(RE,1) = CR(RE,1) + XN(N1):
               CR(RE,2) = CR(RE,2) + YN(N1):
               CR(RE,0) = CR(RE,0) + 1:
     NEXT N,EL

174  FOR RE = 1 TO NQ:
          CR(RE,1) = FN PX(CR(RE,1) / CR(RE,0)):
          CR(RE,2) = FN PY(CR(RE,2) / CR(RE,0)):
     NEXT RE:
     FOR I = 1 TO 2 * NS:
          IS(I) = BN%(I,0):
     NEXT I:
     RETURN
```

Define scale for plot.

```
176  A = (SB - ST) / (VH - VL):
     SC = (SR - SL) / (UH - UL):
     SC = SC * (SC < = A) + A * (SC >A):
     RETURN
```

(178-190) Retrieve node numbers (RE=5), nodal coordinates (RE=6), unique line nodes (RE=7) and boundary nodes (RE=8).

```
178  RE = 5:
     GOSUB 202:
     GOSUB 196:
     ON SI%(RE) = 0 GOTO 192:
     NE = N1:
     DE = N2:
     & C(ND%):
     DIM ND%(NE,DE):
     ONERR  GOTO 218

180  EF = 0:
     GOSUB 206:
     ON BC% GOTO 18,18,178:
     & R(ND%,N$):
```

```
       ON EF GOTO 180:
       & O(ND%,1 TO N1,0,A):
       NQ = ND%(N1,0):
       & R(ND%,N$):
       GOSUB 212:
       RE = 6:
       GOSUB 196:
       ON SI%(RE) = 0 GOTO 192:
       NN = N1:
       NP = 2 * NN:
       & C(XY,XN,YN):
       DIM XY(NN,2),XN(NN),YN(NN)

182    EF = 0:
       GOSUB 206:
       ON BC% GOTO 178,18,178:
       & R(XY,N$):
       ON EF GOTO 182:
       GOSUB 212

184    FOR I = 1 TO NN:
          XN(I) = XY(I,1):
          YN(I) = XY(I,2):
       NEXT I:
       & O(XY,1 TO NN,1,A):
       XL = XY(1,1):
       XH = XY(NN,1):
       & O(XY,1 TO NN,2,A):
       YL = XY(1,2):
       YH = XY(NN,2):
       & C(XY)

186    RE = 7:
       GOSUB 196:
       ON SI%(RE) = 0 GOTO 192:
       NL = N1:
       & C(LN%):
       DIM LN%(NL,3)

188    EF = 0:
       GOSUB 206:
       ON BC% GOTO 178,18,186:
       & R(LN%,N$):
       ON EF GOTO 188:
       GOSUB 212:
       RE = 8:
       GOSUB 196:
       ON SI%(RE) = 0 GOTO 192:
       NB = N1:
       DB = 2 * NB:
       & C(BN%):
       DIM BN%(NB,2)

190    EF = 0:
       GOSUB 206:
       ON BC% GOTO 186,18,186:
       & R(BN%,N$):
       ON EF GOTO 190:
       GOTO 212
```

Warn of invalid data file.

```
192    FLASH :
       ON  PEEK (222) = 77 GOTO 194:
```

```
PRINT "[G][G][M]DATA NOT DEFINED":
NORMAL :
PRINT "EITHER FILE":
INVERSE :
PRINT NA$:
NORMAL :
PRINT "DOESN'T EXIST OR THE DATA IS NOT VALID":
GOSUB 4:
BC% = 2:
RETURN
```

Out of memory error message.

```
194 PRINT "[G][G]SORRY, OUT OF MEMORY":
    NORMAL :
    END
```

Read information about file referenced by record RE.

```
196 N = FRE (0):
    PRINT D$"OPEN"FI$",L100"DR$(DR):
    PRINT D$"READ"FI$",R"RE:
    INPUT SI%(RE),NA$,DE$,N1,D1$,N2,D2$:
    PRINT D$"CLOSE"FI$:
    N$ = NA$ + DR$(DR):
    RETURN
```

Request filename for saving data.

```
198 ON Z = 2 OR PEEK (2163) = 3 GOTO 208:
    PRINT "[M]*** ";:
    INVERSE :
    PRINT "SAVING";:
    NORMAL :
    PRINT " DATA TO DISKETTE ***[M][M]":
    ON Z GOTO 208:
    & I("[M][M]FILE NAME FOR[M]"
       + DE$,NA$,NA$,"24",BC%,30):
    ON BC% >0 GOTO 198:
    GOSUB 210:
    GOTO 208
```

Set active status and save information about record RE.

```
200 SI%(RE) = 1:
    ON PEEK (2163) = 3 GOTO 18:
    PRINT D$"OPEN"FI$",L100"DR$(DR):
    PRINT D$"WRITE"FI$",R"RE:
    PRINT SI%(RE):
    PRINT NA$:
    PRINT DE$:
    PRINT N1:
    PRINT D1$:
    PRINT N2:
    PRINT D2$:
    PRINT D$"CLOSE"FI$:
    PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2" "D2$"[M]
       WERE SAVED":
    RETURN
```

Verify existence of data file.

```
202 ONERR GOTO 226

204 E3 = 0:
    PRINT D$"CKFILE"FI$DR$(DR):
    ON E3 GOTO 204:
    RETURN
```

(206-210) Request filename for data retrieval.

```
206 PRINT "[M]*** ";:
    INVERSE :
    PRINT "LOADING";:
    NORMAL :
    PRINT " DATA FROM DISKETTE ***[M][M]":
    ON Z GOTO 208:
    & I("FILE NAME FOR[M]" + DE$,NA$,NA$,"25",BC%,30):
    ON BC% >0 GOTO 206:
    GOSUB 210

208 PRINT " "NA$:
    N$ = NA$ + DR$(DR):
    RETURN

210 ON LEFT$ (NA$, LEN (KW$)) = KW$ GOTO 18:
    NA$ = KW$ + "/" + NA$:
    RETURN
```

Print file descriptors.

```
212 ON Z = 2 GOTO 18:
    PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2" "D2$"[M]
       WERE LOADED":
    RETURN
```

Read file information.

```
214 PRINT D$"OPEN"FI$",L100"DR$(DR):
    PRINT D$"READ"FI$",R0":
    INPUT TE,KW$,PD$:
    PRINT D$"CLOSE"FI$:
    PRINT "[M]PROBLEM KEYWORD: "KW$"[M][M]
       PROBLEM DESCRIPTION:[M]"PD$:
    RETURN
```

Disable data files.

```
216 ON PEEK (2163) = 3 GOTO 18:
    PRINT D$"OPEN"FI$",L100"DR$(DR):
    FOR RE = R1 TO 24:
        PRINT D$"WRITE"FI$",R"RE:
        PRINT 0:
        SI%(RE) = 0:
    NEXT RE:
    PRINT D$"CLOSE"FI$:
    RETURN
```

(218-226) Trap errors.

```
218 EF = 1:
    POKE 216,0:
    ER = PEEK (222):
    ON ER = 255 GOTO 222:
    ON ER = 77 GOTO 194:
    FLASH :
    PRINT "[G][G][M]ERROR #"ER"[M]":
    NORMAL :
    ER = ER * (ER >3 AND ER <11)
       + NOT (ER >3 AND ER <11):
    ON ER = 1 OR ER = 5 GOTO 220:
    ON ER >3 AND ER <11 GOTO 224

220 GOSUB 196:
    RESUME

222 END
```

```
224  PRINT ER$(ER - 4):
     ON ER = 6 GOSUB 44:
     ON ER = 4 OR ER = 5 OR ER = 8 OR ER = 9 GOSUB 4:
     ON ER <10 GOTO 220:
     & I("UNLOCK? (Y/N)";"Y";A$,"26",0,1):
     ON A$ = "N" GOTO 220:
     PRINT D$"UNLOCK "NA$DR$(DR):
     GOTO 220

226  E3 = 1:
     POKE 216,0:
     ON  PEEK (222) = 255 GOTO 222:
     & I("[G][G][M]DATA DISKETTE IS NOT IN DRIVE[M]"
        +  WR$;"";A$,"27"):
     RESUME
```

(1,228-254) Cold start initialization.

```
228  & D(T):
     POKE 2164,3:
     HOME :
     INVERSE :
     FOR I = 1 TO 4:
        PRINT  TAB( 2)" " TAB( 38)" ":
     NEXT I:
     NORMAL :
     VTAB 2:
     HTAB 3:
     PRINT " APPLIED FINITE ELEMENT ANALYSIS: ":
     HTAB 3:
     PRINT "     AN APPLE " CHR$ (221) CHR$ (219);
       " IMPLEMENTATION     ":
     VTAB 6:
     HTAB 12:
     PRINT "COPYRIGHT, 1985[M]"

230  HTAB 11:
     INVERSE :
     PRINT "PREPROCESS";:
     NORMAL :
     PRINT " (3.00)[M][M]ABSTRACT:[M]
        SPECIFY 'MATERIAL' PROPERTIES AND BOUND-ARY CONDITIONS,
        IN THAT ORDER.[M]THE REQUIRED PROPERTIES ARE THE
        ELASTIC MODULUS, POISSON'S RATIO, THICKNESS,[M]
        TEMPERATURE AND ";

232  PRINT "THERMAL EXPANSION[M]
        COEFFICIENT FOR EACH   ELEMENT.[M][M]
        BOUNDARYCONDITIONS INCLUDE POINT FORCE,
        FIXED   NODAL DISPLACEMENT, AND SURFACE STRESS.":
     GOSUB 6

234  & C(DR$,ER$):
     DIM DR$(2),ER$(6):
     ER$(0) = "DISK IS WRITE PROTECTED":
     ER$(2) = "FILE NOT FOUND":
     ER$(3) = "VOLUME MISMATCH":
     ER$(4) = "I/O ERROR":
     ER$(5) = "DISK IS FULL":
     ER$(6) = "FILE IS LOCKED":
     DR$(1) = ",D1":
     DR$(2) = ",D2":
     DR =  PEEK (2048):
     ON DR = 2 GOTO 238
```

```
236  & B(200,20):
     PRINT "INSERT ";:
     FLASH :
     PRINT "DATA";:
     NORMAL :
     & I(" DISKETTE INTO DRIVE <RETURN>";"";A$,"28"):
     HOME

238  Z =  PEEK (2163):
     GOSUB 202:
     HOME :
     GOSUB 214:
     PRINT :
     POKE 34,7

240  & C(UN$):
     DIM UN$(1):
     UN$(0) = "NOT ":
     DEF  FN P2(I) = I +  1 - I * (I = DE):
     DEF  FN M(B) = ((N <I2) - (I2 - I1) * (N = I2))
        * (A = 21) - ((N >I1) - (I2 - I1)
        * (N = I1)) * (A = 8):
     P2 = 8 *  ATN (1):
     DEF  FN N(N) = I1 * (N <I1 OR N >I2)
        +  N * (N > = I1 AND N < = I2):
     DIM SI%(25),MU$(8):
     TF$ = KW$ +  "/TEMPFILE"

242  MU$(1) = "ENTER MATERIAL PROPERTIES":
     MU$(2) ="ENTER BOUNDARY CONDITIONS":
     MU$(3) = "LIST MATERIAL PROPERTIES":
     MU$(4) = "PLOT MATERIAL PROPERTIES":
     MU$(5) = "LIST INPUT BOUNDARY CONDITIONS":
     MU$(6) = "LIST EQUIV BOUNDARY CONDITIONS"

244  MU$(7) = "PLOT" +  RIGHT$ (MU$(5),26):
     MU$(8) = "PLOT" +  RIGHT$ (MU$(6),26):
     DEF  FN CX(EL) =  FN PX((XN(ND%(EL,1))
        +  XN(ND%(EL,2)) +  XN(ND%(EL,3))) / 3):
     DEF  FN CY(EL) =  FN PY((YN(ND%(EL,1))
        +  YN(ND%(EL,2)) + YN(ND%(EL,3))) / 3)

246  DIM TY$(5),CV(6,5),CO%(9),OP%(9),BT$(3),DF$(4),
        DI$(2),AC$(2),BD$(4):
     DATA 0,0,1,0,1,1,8,0,8,1,16,1,5,1,19,0,10,1,20,0:
     FOR K = 0 TO 9:
        READ CO%(K),OP%(K):
     NEXT K:
     GOSUB 178:
     ON BC% >0 GOTO 228:
     UL = XL:
     UH = XH:
     VL = YL:
     VH = YH

248  FOR I = 1 TO 6:
        CV(I,1) = 1:
        CV(I,2) = .25:
        CV(I,3) = 1:
     NEXT I:
     ST = 20:
     SB = 140:
     SL = 30:
     SR = 250:
```

```
        GOSUB 176:
        DEF  FN PX(X) = SL +  SC * (X - UL):
        DEF  FN PY(Y) = SB - SC * (Y - VL):
        DEF  FN UX(X) = UL +  (X - SL) / SC:
        DEF  FN VY(Y) = VL - (Y - SB) / SC

250  GOSUB 172:
     DIM IO$(2):
     IO$(1) = "INPUT REGION":
     IO$(2) = "ELEMENT"

252  TY$(1) = "ELASTIC MODULUS":
     TY$(2) = "POISSON'S RATIO":
     TY$(3) = "THICKNESS":
     TY$(4) = "TEMPERATURE":
     TY$(5) = "THERMAL EXPANSION COEFF":
     BT$(0) = "FREE BOUNDARY":
     BT$(1) = "POINT FORCE":
     BT$(2) = "DISPLACEMENT":
     BT$(3) = "SURFACE STRESS"

254  DF$(1) = "X":
     DF$(2) = "Y":
     DF$(3) = "R":
     DF$(4) = "Z":
     DI$(1) = DF$(2 * TE - 1):
     DI$(2) = DF$(2 * TE):
     AC$(1) = "ENTER":
     AC$(2) = "DELETE":
     BD$(1) = DI$(1) +  "-DIRECTION":
     BD$(2) = DI$(2) +  "-DIRECTION":
     BD$(3) = "NORMAL (OUT)":
     BD$(4) = "TANGENT (CCW)"
```

(256-258) Warm restart and menu.

```
256  NM =  PEEK (2166):
     TEXT :
     HOME :
     ON Z = 2 AND NM = 3 GOTO 260:
     PRINT
        "** PROPERTIES & BOUND. COND. (3.05) **[M][M]
        --- DATA INPUT --[M]
        1. "MU$(1)"[M]
        2. "MU$(2)"[M][M]
        --- DATA REVIEW --[M]
        3. "MU$(3)"[M]
        4. "MU$(4)"[M]
        5. "MU$(5)"[M]
        6. "MU$(6)"[M]
        7. "MU$(7)"[M]
        8. "MU$(8)"[M][M]
        0. NONE"

258  & D(T):
     & I(SR$,NM;MO,"29",BC%,1,MO > = 0 AND MO <9):
     ON BC% GOTO 228,260,256:
     ON MO = 0 GOTO 260:
     ON MO GOSUB 274,280:
     ON MO >2 GOSUB 292:
     GOTO 256
```

(260-268) Exit.

```
260  POKE 2166,0:
```

```
     HOME :
     PRINT
        "***      PREPROCESS: EXIT  (3.9)      ***[M]
        1. PROCEED TO 'SOLVE' PROGRAM[M]
        2. EXIT TO MAIN MENU[M]
        3. EXIT TO 'CREATE TEXT FILE'[M]
        4. DON'T EXIT; REMAIN IN 'PREPROCESS'[M][M]
        0. STOP[M]"

262  & I(SR$,1;EO,"30",BC%,1,EO > = 0 AND EO <5):
     ON EO = 4 GOTO 256:
     IF EO = 0 OR BC% = 2 THEN
          END

264  POKE 2166,1:
     & C(PN$):
     DIM PN$(3):
     PN$(1) = "SOLVE":
     PN$(2) = "HELLO":
     PN$(3) = "CREATE TEXT FILE"
```

Request program diskette if one drive or creating a text file of the problem formulation.

```
266  IF DR = 1 OR EO = 3 THEN
          & B(200,20):
          PRINT "[M]INSERT ";:
          FLASH :
          PRINT "PROGRAM";:
          NORMAL :
          PRINT " DISKETTE CONTAINING[M]' ";:
          INVERSE :
          PRINT PN$(EO);:
          NORMAL :
          & I(" ' INTO DRIVE 1[M]" +  WR$;"";A$,"31",BC%):
          ON BC%    0 GOTO 260
```

Adjust the location of the Applesoft program.

```
268  POKE 103, PEEK (103) * (EO < >1) +  63 * (EO = 1):
     POKE 104, PEEK (104) * (EO < >1) +  25 * (EO = 1):
     POKE 6462, PEEK (6462) * (EO < >1) +  0 * (EO = 1):
     PRINT D$"RUN "PN$(EO)",D1"
```

Retrieve material properties (RE=9).

```
270  RE = 9:
     GOSUB 202:
     GOSUB 196:
     ON SI%(RE) = 0 GOTO 192:
     ONERR  GOTO 218

272  EF = 0:
     GOSUB 206:
     & C(MP):
     DIM MP(N1,N2):
     & R(MP,N$):
     ON EF GOTO 272:
     GOSUB 212:
     GOTO 6
```

Menu option 1: Enter material properties.

```
274  TEXT :
     & D(T):
     HOME :
     PRINT "* "MU$(1)" (3.1) *[M]":
     POKE 34,2:
```

```
       & I("RETRIEVE PROPERTIES?";"N";A$,"32",BC%,1):
       ON BC% >0 GOTO 18:
       & C(MP,RP):
       DIM MP(NE,5),RP(NQ,5):
       ED = (A$ = "Y"):
       ON ED + 1 GOSUB 332,270:
       ON BC% >0 GOTO 18:
       GOSUB 70:
       RE = 9:
       GOSUB 196:
       N1 = NE:
       N2 = 5:
       ONERR GOTO 218
```

Save material properties (RE=9) and also save the thickness in a temporary file (as a memory saving technique).

```
276  EF = 0:
     GOSUB 198:
     ON PEEK (2163) = 3 GOTO 278:
     & S(MP,N$):
     PRINT D$"OPEN"TF$;DR$(DR):
     PRINT D$"DELETE"TF$:
     PRINT D$"OPEN"TF$:
     PRINT D$"WRITE"TF$:
     FOR I = 0 TO NE:
         PRINT MP(I,3):
     NEXT I:
     PRINT D$"CLOSE"TF$:
     ON EF GOTO 276

278  GOSUB 200:
     R1 = 11:
     GOSUB 216:
     & C(MP,RP):
     POKE 2166,2:
     & I("[M][M]PROCEED TO BOUNDARY CONDITIONS","Y";
        A$,"33",0,1):
     ON A$ = "N" GOTO 18:
     MO = 2
```

Menu option 2: Enter boundary conditions.

```
280  TEXT :
     & D(T):
     HOME :
     PRINT "* "MU$(2)" (3.2) *[M]":
     POKE 34,2:
     RE = 9:
     GOSUB 202:
     GOSUB 196:
     ON SI%(9) = 0 GOTO 282:
     & I("RETRIEVE BOUNDARY CONDITIONS? (Y/N)","N";A$,
        "34",BC%,1):
     ON BC% >0 GOTO 18:
     ED = (A$ = "Y"):
     GOTO 290

282  PRINT "[G][G][M]MATERIAL PROPERTIES MUST BE
        DEFINED    BEFORE BOUNDARY CONDITIONS.":
     GOTO 4

284  RE = 10:
     GOSUB 196:
     ON SI%(RE) = 0 GOTO 192:
     NI = N1:
```

```
    & C(BI):
    DIM BI(NI,4):
    ONERR  GOTO 218

286  EF = 0:
    GOSUB 206:
    & R(BI,N$):
    ON EF GOTO 286:
    GOSUB 212:
    GOTO 6

288  & C(BI%,BV):
    DIM BI%(DB,3),BV(DB):
    FOR I = 1 TO NI:
        BV(I) = BI(I,4):
        FOR J = 1 TO 3:
            BI%(I,J) = BI(I,J):
    NEXT J,I:
    & C(BI):
    RETURN

290  & C(BI%,BV):
    DIM BI%(DB,3),BV(DB):
    ON ED GOSUB 284:
    ON BC% >0 GOTO 18:
    ON ED GOSUB 288:
    & C(TH):
    DIM TH(NE):
    PRINT D$"OPEN"TF$DR$(DR):
    PRINT D$"READ"TF$:
    FOR I = 0 TO NE:
        INPUT TH(I):
    NEXT I:
    PRINT D$"CLOSE"TF$:
    GOSUB 128:
    & C(BI%,BV):
    ON BC% >0 GOTO 18:
    POKE 2165,4:
    POKE 2166,3:
    GOTO 4
```

Data review options (3-8).

```
292  TEXT :
    & D(T):
    HOME :
    PRINT "* "MU$(MO)" (3."MO") *[M]":
    POKE 34,2:
    ON MO - 2 GOSUB 270,270,284,318,284,318:
    ON BC% >0 GOTO 18:
    ON MO - 2 GOTO 294,300,314,322,326,328
```

(294-298) Menu option 3: List the properties.

```
294  & L(P%):
    FOR TY = 1 TO 5:
        TY = TY + (TY = 3 AND TE = 2):
        POKE 34,2:
        HOME :
        PRINT  TAB( (40 -  LEN (TY$(TY)) - 10) / 2)
           "PROPERTY: ";:
        INVERSE :
        PRINT TY$(TY)"[M]":
        IF MP(0,TY) < >1 THEN
            PRINT "ELEMENT "TY$(TY):
          POKE 34,5:
```

```
            HOME :
            VTAB 6:
            NORMAL :
            FOR EL = 1 TO NE:
                PRINT "  "EL TAB( 12)MP(EL,TY):
        NEXT EL

296     IF MP(0,TY) = 1 THEN
             PRINT TY$(TY)" IS UNIFORM.":
            NORMAL :
            PRINT "[M]FOR ALL ELEMENTS (1 TO "NE") THE[M]
               "TY$(TY)" VALUE IS ";:
            INVERSE :
            PRINT MP(1,TY):
            NORMAL

298     GOSUB 4:
        TY = TY - (BC% = 1) +  5 * (BC% = 2):
    NEXT TY:
    & L(255):
    & C(MP):
    POKE 2166,4:
    RETURN
```

(300-312) Menu option 4: Plot selected properties.

```
300 HOME :
    PRINT "SELECT PROPERTY TO PLOT[M][M]":
    FOR I = 1 TO 5:
        PRINT I". "TY$(I):
    NEXT I:
    PRINT "[M]6. ALL OF THESE[M][M]0. NONE[M][M]":
    & I(SR$,0;TY,"35",BC%,1,(TY >
       = 0 AND TY <7) AND  NOT (TY = 3 AND TE = 2)):
    ON BC% < >0 OR TY = 0 GOTO 18:
    T1 = (TY = 6) +  TY * (TY < >6):
    T2 = 5 * (TY = 6) +  TY * (TY < >6)

302 FOR TY = T1 TO T2:
        ON  NOT (TY = 3 AND TE = 2) GOSUB 304:
        TY = TY +  6 * (BC% = 2) - (2 * (TY >1)
           +  (TY = 1)) * BC% = 1):
    NEXT TY:
    & C(MP,PV):
    POKE 2166,5:
    RETURN

304 POKE 34,20:
    HOME :
    PRINT  TAB( (40 -  LEN (TY$(TY)) - 10) / 2)"PROPERTY: ";:
    INVERSE :
    PRINT TY$(TY):
    NORMAL :
    POKE 34,21:
    HOME :
    ON MP(0,TY) GOTO 312:
    HGR :
    & D(G):
    HCOLOR= 3:
    & C(PV):
    DIM PV(9)

306 FOR I = 1 TO NL:
        N1 = LN%(I,1):
        N2 = LN%(I,2):
```

```
       HPLOT  FN PX(XN(N1)), FN PY(YN(N1))
          TO  FN PX(XN(N2)), FN PY(YN(N2)):
    NEXT I:
    J = 1:
    NC = 0:
    & O(MP,1 TO NE,TY,A)

308 NC = NC +  1:
    PV(NC) = MP(J,TY):
    & W((MP,1 TO NE,TY),PV(NC),PV(NC),N1%,BR%,ER%):
    PRINT :
    FOR I = BR% TO ER%:
        EL = MP(I,0):
        & F(CO%(NC),OP%(NC), FN CX(EL), FN CY(EL)):
    NEXT I:
    ON ER% = NE GOTO 310:
    J = ER% +  1:
    GOTO 308

310 HOME :
    POKE 34,0:
    FOR I = 1 TO NC:
        VTAB 22 +  (I >3) +  (I >6):
        HTAB ((I - 6) * (I >6) +  (I - 3)
           * (I >3 AND I <7) +  I * (I <4))
           * 12 - 10:
        PRINT  CHR$ (64 +  I)"="PV(I):
    NEXT I:
    PRINT "(A=DENSE TO " CHR$ (64 +  NC)"=SPARSE DOTS)";:
    & I("  <RETURN>";"";A$,"36",BC%):
    RETURN

312 GOSUB 56:
    & F(1,1, FN CX(1), FN CY(1)):
    PRINT "[M]"TY$(TY)" IS UNIFORM.[M]
       FOR ALL ELEMENTS (1 TO "NE") THE[M]VALUE IS ";:
    INVERSE :
    PRINT MP(1,TY);:
    NORMAL :
    & I("  <RETURN>";"";A$,"37",BC%):
    RETURN
```

(314-320) Menu option 5: List the user-specified boundary conditions.

```
314 PRINT "INPUT B.C. SPECIFICATIONS:":
    INVERSE :
    PRINT "NODE   B.C. DIRECTION & TYPE     VALUE[M]":
    NORMAL :
    POKE 34,5

316 & L(P%):
    FOR I = 1 TO NI:
        PRINT BN%(BI(I,1),1) TAB( 5)BD$(BI(I,3))" "
           BT$(BI(I,2))" " TAB( 34)BI(I,4):
    NEXT I:
    & L(255):
    TEXT :
    & D(T):
    POKE 2166,6:
    & C(BI):
    GOTO 4

318 RE = 11:
    GOSUB 202:
    GOSUB 196:
```

```
        ON SI%(RE) = 0 GOTO 192:
        & C(BC):
        DIM BC(NP,2):
        ONERR  GOTO 218

320  EF = 0:
        GOSUB 206:
        & R(BC,N$):
        ON EF GOTO 320:
        GOSUB 212:
        GOTO 6
```

(322-324) Menu option 6: List the nodal equivalent boundary conditions.

```
322  HOME :
        PRINT "CONDITIONS AT BOUNDARY NODES:":
        INVERSE :
        PRINT "NODE" TAB( 10)"B.C. TYPE" TAB( 25)"B.C. VALUE[M]":
        NORMAL :
        POKE 34,5

324  & L(P%):
        FOR I = 1 TO NB:
            FOR J = 1 TO 2:
                K = 2 * BN%(I,1) - 2 + J:
                PRINT " "BN%(I,1)DI$(J) TAB( 8)BT$(BC(K,1))
                   TAB( 25)BC(K,2):
        NEXT J,I:
        & L(255):
        TEXT :
        & D(T):
        POKE 2166,7:
        & C(BC):
        GOTO 4
```

Menu option 7: Plot the user-specified boundary condtions.

```
326  HOME :
        VTAB 22:
        PRINT  TAB( 8)"INPUT BOUNDARY CONDITIONS":
        GOSUB 288:
        GOSUB 56:
        GOSUB 64:
        POKE 2166,8:
        & C(BI%,BV):
        GOTO 4
```

(328-332) Menu option 8: Plot the nodal equivalent boundary conditions.

```
328  VTAB 22:
        PRINT  TAB( 8)"EQUIV. BOUNDARY CONDITIONS":
        GOSUB 56:
        SZ = 0:
        FOR N = 1 TO NB:
            NO = BN%(N,1):
            B = 0:
            GOSUB 62

330         A = 8:
            B1 = N +  FN M(B):
            A = 21:
            B2 = N +  FN M(B):
            N1% = BN%(B1,1):
            N2% = BN%(B2,1):
            GOSUB 8:
                FOR DI = 1 TO 2:
                    DF = 2 * NO - 2 +  DI:
```

```
            BT = BC(DF,1):
            VB = BC(DF,2):
            ROT= 8 * (DI = 2):
            SCALE= 2 +  SZ:
            SY = 64:
            ON BT < >0 AND
               NOT (BT = 1 AND VB = 0) GOSUB 48:
      NEXT DI,N:
      SZ = 1:
      & C(BC):
      GOTO 4

332  FOR I = 1 TO NE:
         MP(I,1) = 1:
         MP(I,2) = .25:
         MP(I,3) = 1:
     NEXT :
     RETURN

65535 REM [M][M]1FEB85[M]JRC/DCD

END-OF-LISTING
```

SOLVE.ELASTICITY

(1,212-228) Cold start initialization.

```
1  CLEAR :
   TEXT :
   & D(T):
   P% = 10:
   & L(255):
   SR$ = "... SELECT BY NUMBER <RETURN>":
   WR$ = "... WHEN READY, PRESS <RETURN>":
   D$ =  CHR$ (4):
   FI$ = "FILEINFO.TXT":
   POKE 2164,4:
   Z =  PEEK (2163):
   GOTO 212:
   REM SOLVE(E)
```

(2,230-236) Warm restart and menu.

```
2  GOTO 230
```

Wait for user response, clear screen, and continue.

```
4  ON FR = 1 GOSUB 244:
   VTAB (FR = 1) +  24 * (FR = 0):
   CALL  - 958:
   ON FR = 1 GOTO 18:
   & B(20,20):
   & I(WR$;"";A$,"1",BC%):
   HOME :
   RETURN
```

(6-10) Display options during assembly of global matrices.

```
6  ON MO < >2 GOTO 4:
   & B(20,20):
   & I("[M]<D>CHANGES DISPLAY OPTION, <RETURN>";
     "";A$,"2",BC%,1):
   HOME :
   ON A$ < >"D" GOTO 18:
   POKE 34,2:
   HOME

8  PRINT "[M]* DISPLAY OPTIONS *[M][M]
     1. ONLY FINAL GLOBAL MATRICES[M]
     2. ELEMENT & GLOBAL MATRICES[M]
     3. DETAIL DURING CALCULATION[M][M]
     0. NO DISPLAY[M]":
   & I(SR$,1;PO,"3",BC%,1,PO > = 0 AND PO <4):
   POKE  - 16368,0:
   POKE 34,2:
   HOME :
   IF PO >1 OR EL = 0 THEN
        POKE 35,24:
      HOME :
      RETURN

10 ON FR GOTO 18:
   VTAB 20:
   PRINT "  PRESS <ESC>TO CHANGE DISPLAY OPTION":
   RETURN
```

Display moving '' and check for 'esc' keypress.*

```
12  POKE 35,24:
    INVERSE :
    VTAB 12 +  V:
    HTAB 19 +  H:
    PRINT " ";
    H1 = (H = V):
    H = H *  NOT H1 +   NOT H * H1:
    V = V * H1 +   NOT V *  NOT H1:
    VTAB 12 +  V:
    HTAB 19 +  H:
    PRINT "*";
    NORMAL :
    ON  PEEK ( - 16384) < = 127 GOTO 18:
    POKE 34,2:
    HOME :
    GET A$:
    GOTO 8
```

(14-18) Decompose global equations.

```
14  FOR I = 1 TO NP - 1:
        VTAB 5:
        PRINT "  WORKING ON ROW #"I" OF "NP:
        G1 = GS(I,1):
        MJ = I +  BW - 1:
        MJ = MJ * (MJ < = NP) +  NP * (MJ >NP):
        N = 0:
        MK = BW:
        IF NP - I +  1 <BW THEN
            MK = NP - I +  1


16      FOR J = I +  1 TO MJ:
            GOSUB 12:
            MK = MK - 1:
            N = N +  1:
            G2 = GS(I,N +  1):
            FOR K = 1 TO MK:
                GS(J,K) = GS(J,K) - G2 * GS(I,N +  K) / G1:
            NEXT K:
            GF(J) = GF(J) - G2 * GF(I) / G1:
        NEXT J:
        PRINT :
    NEXT I

18  RETURN
```

(20-26) Modify global equations as required by constant displacement boundary conditions.

```
20  FOR DF = 1 TO NP:
        VTAB 6:
        HTAB 5:
        PRINT "WORKING ON ROW "DF" OF "NP:
        ON BC(DF,1) < >2 GOTO 26:
        BV = BC(DF,2):
        K = DF - 1:
        FOR J = 2 TO BW:
            M = DF +  J - 1:
            IF M < = NP THEN
                GF(M) = GF(M) - GS(DF,J) * BV:
                GS(DF,J) = 0

22          IF K >0 THEN
                GF(K) = GF(K) - GS(K,J) * BV:
```

```
                GS(K,J) = 0:
                K = K - 1

24          NEXT J:
            GS(DF,1) = GS(DF,1) * (GS(DF,1) > = .0005)
                + 50000000 * (GS(DF,1) <.0005):
            GF(DF) = GS(DF,1) * BV

26      NEXT DF:
        RETURN
```

(28-32) Define material property (elasticity) matrix for element EL.

```
28      & M(D = (0)):
        IF PS = 1 AND TE = 1 THEN
            R = MP(EL,1) / (1 - MP(EL,2) ^ 2):
            D(1,2) = MP(EL,2) * R:
            D(3,3) = R * (1 - MP(EL,2)) / 2

30      IF (PS = 2 AND TE = 1) OR TE = 2 THEN
            R = MP(EL,1) * (1 - MP(EL,2)) / ((1 + MP(EL,2))
                * (1 - 2 * MP(EL,2))):
            FOR I = 1 TO DD - 2:
                FOR J = I + 1 TO DD - 1:
                    D(I,J) = R * MP(EL,2) / (1 - MP(EL,2)):
            NEXT J,I:
        D(DD,DD) = R * (1 - 2 * MP(EL,2))
            / (2 * (1 - MP(EL,2)))

32      FOR I = 1 TO DD - 1:
            D(I,I) = R:
        NEXT I:
        FOR I = 1 TO DD - 2:
            FOR J = I + 1 TO DD - 1:
                D(J,I) = D(I,J):
        NEXT J,I:
        ON PO <3 GOTO 18:
        & L(0):
        PRINT "[M]MATERIAL ELASTICITY MATRIX":
        FOR I = 1 TO DD:
            INVERSE :
            PRINT "[M]ROW "I TAB( 39)" ":
            NORMAL :
            FOR J = 1 TO DD:
                PRINT "   "D(I,J);:
        NEXT J,I:
        PRINT :
        & L(255):
        GOTO 6
```

(34-42) Find element centroid, element area, initial thermal strain, and shape function gradient matrix for element EL.

```
34      RC = 0:
        ZC = 0:
        FOR I = 1 TO DE:
            ND(I) = EN%(EL,I):
            X(I) = NC(ND(I),1):
            Y(I) = NC(ND(I),2):
            NS(2 * I - 1) = 2 * ND(I) - 1:
            NS(2 * I) = 2 * ND(I):
            RC = RC + X(I):
            ZC = ZC + Y(I):
        NEXT I:
        RC = RC / DE:
        ZC = ZC / DE
```

```
36  DT = MP(EL,4) - MP(0,0):
    A2 = X(2) * Y(3) +  X(3) * Y(1) +  X(1) * Y(2) - X(2)
       * Y(1) - X(3) * Y(2) - X(1) * Y(3):
    K = (1 +  MP(EL,2) * (TE = 1 AND PS = 2))
       * MP(EL,5) * DT:
    & M(E0 = (K)):
    E0(DD,1) = 0:
    & M(B = (0))

38  FOR J = 1 TO DE:
       I =  FN P1(J):
       K =  FN P2(J):
       B(1,2 * J - 1) = (Y(K) - Y(I)) / A2:
       B(2,2 * J) = (X(I) - X(K)) / A2:
       B(DD,2 * J) = B(1,2 * J - 1):
       B(DD,2 * J - 1) = B(2,2 * J):
       IF TE = 2 THEN
          B(DD - 1,2 * J - 1) = ((X(K) * Y(I) - X(I)
             * Y(K)) / RC +  Y(K) - Y(I) +  (X(I)
             - X(K)) * ZC / RC) / A2

40  NEXT J:
    ON PO <3 GOTO 18:
    & L(0):
    PRINT "[M]ELEMENT NO. "EL"  NODE LOCATIONS":
    FOR I = 1 TO DE:
       PRINT " NODE "ND(I)":   "X(I) TAB( 25)Y(I):
    NEXT I:
    PRINT "[M]TEMPERATURE DIFFERENCE = "DT"[M][M]
       ELEMENT AREA = "A2 / 2:
    GOSUB 6:
    HOME

42  PRINT "SHAPE FUNCTION GRADIENT MATRIX":
    FOR I = 1 TO DD:
       INVERSE :
       PRINT "DNJ/DX"I TAB( 39)" ":
       NORMAL :
       FOR J = 1 TO DK:
          PRINT  CR$(  FN R(B(I,J)))B(I,J);:
       NEXT J:
       PRINT :
    NEXT I:
    & L(255):
    GOTO 6
```

(44-70) Menu option 2: Find initial force and stiffness matrices for each element, assemble the global force and stiffness matrices, and save (RE=12,13).

```
44  POKE 34,0:
    HOME :
    PRINT "FORCE & STIFFNESS MATRICES (4.2)[M][M]":
    POKE 34,2:
    & C(D,ND,X,Y,B,BT,NS,E0,KE,FE,C,GF,GS):
    DIM D(DD,DD),ND(DE),X(DE),Y(DE),B(DD,DK),BT(DK,DD),
       NS(DK),E0(DD,1),KE(DK,DK),FE(DK,1),C(DK,DD),
       GF(NP),GS(NP,BW):
    PO = 0

46  EL = 0:
    ON FR = 0 GOSUB 8:
    H = 0:
    V = 0:
    EL = 1:
    ON UN GOSUB 28
```

1) 2→DD-1

DD-1

DD-1→2

2)

3)

```
48  FOR EL = 1 TO NE:
       POKE 34,2:
       POKE 35,24 - 14 * (PO <2 AND EL >1):
       HOME :
       PRINT "ASSEMBLING ELEMENT "EL" OF "NE" ELEMENTS...":
       IF PO <2 AND EL = 1 THEN
           GOSUB 10

50     ON UN = 0 GOSUB 28:
       GOSUB 34:
       & M(BT = TRN(B)):
       & M(C = BT.D):
       & M(KE = C.B):
       VE = (MP(EL,3) * (TE = 1) +  P2 * RC
         * (TE = 2)) * A2 / 2:
       & M(KE = KE * (VE)):
       & M(FE = C.E0):
       & M(FE = FE * (VE))

52     ON PO <2 GOTO 56:
       & L(0):
       K = K +  DK * (BC% = 2):
       PRINT "[M]STIFFNESS MATRIX FROM ELEMENT "EL"[M]":
       FOR K = 2 TO DK STEP 2:
           INVERSE :
           PRINT  TAB( 10)"COLUMN "NS(K - 1) TAB( 25)
             "COLUMN "NS(K) TAB( 39)" ":
           NORMAL :
           FOR I = K - 1 TO K:
               PRINT "ROW "NS(I) TAB( 10)KE(I,K - 1)
                  TAB( 25)KE(I,K):
       NEXT I,K

54     GOSUB 6:
       ON PO <2 GOTO 56:
       INVERSE :
       PRINT "[M][M]ELEMENT "EL" THERMAL FORCE VECTOR[M]":
       NORMAL :
       FOR I = 1 TO DK:
           PRINT "ROW "NS(I) TAB( 10)FE(I,1):
       NEXT I:
       PRINT :
       GOSUB 6:
       IF PO = 3 THEN
            & L(P%):
           INVERSE :
           PRINT "UPDATES TO GLOBAL STIFFNESS AFTER[M]
              ELEMENT "EL:
           NORMAL :
           POKE 34,4

56     FOR I = 1 TO DK:
           ON PO <2 GOSUB 12:
           KI = NS(I):
           GF(KI) = GF(KI) +  FE(I,1):
           FOR J = 1 TO DK:
               KJ = NS(J):
               IF KJ > = KI THEN
                   K = KJ - KI +  1:
                   GS(KI,K) = GS(KI,K) +  KE(I,J):
                   IF PO >2 THEN
                        PRINT "K("KI","KJ")" TAB( 10)
                          "IS "GS(KI,K)
```

```
58      NEXT J,I:
        & L(255):
        POKE 34,(2 * (PO >2) +   PEEK (34)
           * (PO <3)):
        ON PO >2 GOSUB 6:
        ON PO >2 GOSUB 156:
        ON PO >2 GOSUB 158:
        IF PO >1 THEN
             PRINT

60  NEXT EL:
    POKE 34,2:
    HOME :
    PRINT "* GLOBAL MATRICES ASSEMBLED *";:
    GOSUB 4:
    IF PO >0 THEN
         GOSUB 156:
         GOSUB 158

62  PRINT :
    POKE 34,2:
    HOME :
    RE = 12:
    GOSUB 180:
    GOSUB 172:
    N1 = NP:
    N2 = 1:
    ONERR  GOTO 200

64  EF = 0:
    GOSUB 174:
    ON  PEEK (2163) = 3 GOTO 66:
    & S(GF,N$)

66  ON EF GOTO 64:
    GOSUB 176:
    RE = 13:
    GOSUB 172:
    N1 = NP:
    N2 = BW

68  EF = 0:
    GOSUB 174:
    ON  PEEK (2163) = 3 GOTO 70:
    & S(GS,N$)

70  ON EF GOTO 68:
    GOSUB 176:
    R1 = 14:
    R2 = 24:
    GOSUB 198:
    POKE 34,0:
    POKE 2166,3:
    GOTO 4
```

(72-92) Menu option 3: Apply the force boundary conditions.

```
72  POKE 34,0:
    HOME :
    PRINT "* COMB. BOUND. & INIT. FORCES (4.3 ) *[M][M]":
    POKE 34,2:
    RE = 12:
    GOSUB 180:
    GOSUB 172:
    ON SI%(RE) GOTO 76:
```

```
        GOSUB 74:
        ON BC% GOTO 44,76

74  PRINT "[G][G][M]GLOBAL FORCE AND STIFFNESS MATRICES[M]
        MUST BE ASSEMBLED BEFORE APPLICATION OF
        EXTERNAL FORCE BOUNDARY CONDITIONS[M]":
    & I("DEFINE NOW";"Y";A$,"4",0,1):
    BC% = 1 + (A$ = "N"):
    RETURN

76  ONERR GOTO 200

78  EF = 0:
    GOSUB 184:
    & R(GF,N$):
    ON EF GOTO 78:
    GOSUB 190:
    NP = N1:
    GOSUB 4

80  HOME :
    PO = 0:
    ON FR GOTO 82:
    PRINT "DISPLAY OPTIONS (4.35)[M][M]
       1. ORIGINAL GLOBAL FORCE VECTOR[M]
       2. NEW GLOBAL FORCE VECTOR[M]
       3. BOTH FORCE VECTORS[M][M]
       0. NO DISPLAY":
    & I(SR$,1;PO,"5",BC%,1,PO > = 0 AND PO <4)

82  LA$ = "------ THERMAL FORCE VECTOR ---------":
    ML = 1:
    ON PO = 1 OR PO = 3 GOSUB 84:
    FOR I = 1 TO NP:
        GF(I) = GF(I) + BC(I,2) * (BC(I,1) = 1):
    NEXT I:
    LA$ = "----- VECTOR INCLUDING FORCE B.C. -----":
    ON PO = 2 OR PO = 3 GOSUB 84:
    GOTO 86

84  HOME :
    PRINT LA$:
    POKE 34,4:
    ON ML GOSUB 156,158:
    RETURN

86  ONERR GOTO 200

88  HOME :
    RE = 14:
    GOSUB 180:
    GOSUB 172:
    N1 = NP:
    N2 = 1:
    ONERR GOTO 200

90  EF = 0:
    GOSUB 174:
    ON PEEK (2163) = 3 GOTO 92:
    & S(GF,N$)

92  ON EF GOTO 90:
    GOSUB 176:
    R1 = 15:
```

```
R2 = 24:
GOSUB 198:
POKE 34,0:
POKE 2166,4:
GOTO 4
```

(94-126) Menu option 4: Apply displacement boundary conditions.

```
94  POKE 34,0:
    HOME :
    PRINT "* APPLY DISPL. BOUNDARY COND. (4.4 ) *[M][M]":
    POKE 34,2:
    RE = 14:
    GOSUB 180:
    GOSUB 172:
    ON SI%(RE) = 1 GOTO 98:
    PRINT "[G][G][M]BOUNDARY AND INITIAL (THERMAL) FORCES[M]
       MUST BE COMBINED BEFORE APPLICATION OF"

96  PRINT "DISPLACEMENT BOUNDARY CONDITIONS[M]":
    & I("COMBINE NOW";"Y";A$,"6",0,1):
    ON A$ = "N" GOTO 98:
    GOTO 72

98  ONERR  GOTO 200

100 EF = 0:
    GOSUB 184:
    ON  PEEK (2163) = 3 GOTO 102:
    & R(GF,N$)

102 ON EF GOTO 100:
    GOSUB 190:
    NP = N1

104 RE = 13:
    GOSUB 180:
    GOSUB 172:
    BC% = 0:
    ON SI%(RE) = 0 GOSUB 74:
    ON BC% = 1 GOTO 72:
    ONERR  GOTO 200

106 EF = 0:
    GOSUB 184:
    ON  PEEK (2163) = 3 GOTO 108:
    & R(GS,N$)

108 ON EF GOTO 106:
    GOSUB 190:
    NP = N1:
    BW = N2:
    GOSUB 4

110 HOME :
    PO = 0:
    ON FR GOTO 116:
    PRINT "DISPLAY OPTIONS (4.45)[M][M]
       1. ORIGINAL STIFFNESS MATRIX[M]
       2. MODIFIED STIFFNESS MATRIX[M]
       3. ORIGINAL FORCE VECTOR[M]
       4. MODIFIED FORCE VECTOR[M]
       5. ALL OF THE ABOVE (OPT 1 TO 4)"

112 PRINT "6. MODIFIED STIFFNESS & FORCE
```

```
     MATRICES    ( OPT 2 & 4 )[M][M]
     0. NO DISPLAY[M]":
   & I(SR$,0;PO,"7",BC%,1,PO > = 0 AND PO <7)

114  ML = 2:
     LA$ = " ------ ORIGINAL MATRIX ------":
     ON PO = 1 OR PO = 5 GOSUB 84:
     ML = 1:
     LA$ = "--- VECTOR AFTER FORCE B.C. ---":
     ON PO = 3 OR PO = 5 GOSUB 84

116  HOME :
     GOSUB 20:
     ML = 1:
     LA$ = "--- VECTOR AFTER DISPLACEMENT B.C. ---":
     ON PO = 4 OR PO = 5 OR PO = 6 GOSUB 84:
     ML = 2:
     LA$ = "--- MATRIX AFTER DISPLACEMENT B.C. ---":
     ON PO = 2 OR PO = 5 OR PO = 6 GOSUB 84

118  POKE 34,2:
     HOME :
     RE = 15:
     GOSUB 180:
     GOSUB 172:
     N1 = NP:
     N2 = 1:
     ONERR  GOTO 200

120  EF = 0:
     GOSUB 174:
     ON  PEEK (2163) = 3 GOTO 122:
     & S(GF,N$)

122  ON EF GOTO 120:
     GOSUB 176:
     RE = 16:
     GOSUB 172:
     N1 = NP:
     N2 = BW

124  EF = 0:
     GOSUB 174:
     ON  PEEK (2163) = 3 GOTO 126:
     & S(GS,N$)

126  ON EF GOTO 124:
     GOSUB 176:
     R1 = 17:
     R2 = 24:
     GOSUB 198:
     POKE 34,0:
     POKE 2166,5:
     GOTO 4
```

(128-148) Menu option 5: Solve for node displacements.

```
128  POKE 34,0:
     HOME :
     PRINT "* SOLVE FOR NODAL DISPLACEMENTS (4.5) *[M][M]":
     POKE 34,2:
     RE = 15:
     ON SI%(RE) = 1 GOTO 138:
     GOSUB 180:
```

```
GOSUB 172:
ON SI%(RE) = 1 GOTO 132

130  PRINT "[G][G][M]DISPLACEMENT BOUNDARY CONDITIONS MUST
        BEAPPLIED BEFORE SOLVING FOR NODAL         DISPLACEMENTS[M]":
     & I("APPLY NOW";"Y";A$,"8",0,1):
     ON A$ = "N" GOTO 132:
     GOTO 94

132  ONERR  GOTO 200

134  EF = 0:
     GOSUB 184:
     & R(GF,N$):
     ON EF GOTO 134:
     GOSUB 190:
     NP = N1:
     RE = 16:
     GOSUB 172:
     ON SI%(RE) = 0 GOTO 130

136  EF = 0:
     GOSUB 184:
     & R(GS,N$):
     ON EF GOTO 136:
     GOSUB 190:
     NP = N1:
     BW = N2:
     GOSUB 4

138  HOME :
     PO = 0:
     IF FR = 0 THEN
        & I("LIST NODE DISPLACEMENTS AS OBTAINED";"N";
           A$,"9",BC%,1):
        PO = (A$ = "Y")

140  HOME :
     PRINT "[M]DECOMPOSING MATRICES...":
     H = 0:
     V = 0:
     GOSUB 14:
     HOME :
     PRINT "SOLVING FOR DISPLACEMENTS...

142  & C(UU):
     DIM UU(NP):
     UU(NP) = GF(NP) / GS(NP,1):
     FOR K = 1 TO NP - 1:
        ON PO = 0 GOSUB 12:
        I = NP - K:
        MJ = BW * (I +  BW - 1 < = NP) +  (NP - I +  1)
           * (I +  BW - 1 >NP):
        SU = 0:
        FOR J = 2 TO MJ:
           N = I +  J - 1:
           SU = SU +  GS(I,J) * UU(N):
        NEXT J:
        UU(I) = (GF(I) - SU) / GS(I,1):
        ON PO = 1 GOSUB 160:
     NEXT K:
     ON PO GOSUB 4:
     HOME
```

```
144  UU(0) = PS:
     RE = 17:
     GOSUB 180:
     GOSUB 172:
     N1 = NP:
     N2 = 1:
     ONERR  GOTO 200

146  EF = 0:
     GOSUB 174:
     ON  PEEK (2163) = 3 GOTO 148:
     & S(UU,N$)

148  ON EF GOTO 146:
     GOSUB 176:
     R1 = 18:
     R2 = 24:
     GOSUB 198:
     POKE 34,0:
     POKE 2166,6:
     GOTO 4
```

(150-154) Menu option 6: List the displacement of nodes.

```
150  POKE 34,0:
     HOME :
     PRINT "  LIST DISPLACEMENTS (4.6)[M]":
     POKE 34,2:
     RE = 17:
     ON SI%(RE) = 1 GOTO 154:
     GOSUB 180:
     & C(UU):
     DIM UU(N1):
     GOSUB 172:
     ON SI%(RE) = 0 GOTO 168:
     ON BC% = 2 GOTO 18:
     ONERR  GOTO 200

152  EF = 0:
     GOSUB 184:
     & R(UU,N$):
     ON EF GOTO 152:
     NP = N1:
     GOSUB 190:
     GOSUB 154:
     POKE 34,0:
     RETURN

154  FR = 0:
     HOME :
     & L(P%):
     INVERSE :
     PRINT  TAB( 11)"NODE DISPLACEMENTS" TAB( 39)" ":
     NORMAL :
     PRINT "[M]NODE   "DI$(1) TAB( 23)DI$(2)"[M]"U$:
     POKE 34,6:
     FOR I = 2 TO NP STEP 2:
        PRINT " "I / 2 TAB( 6)UU(I - 1) TAB( 22)UU(I):
     NEXT I:
     & L(255):
     GOTO 4
```

List the global force vector.

```
156  HOME :
     & L(P%):
```

```
INVERSE :
PRINT  TAB( 10)"GLOBAL FORCE VECTOR" TAB( 39)" [M]":
NORMAL :
POKE 34, PEEK (37):
FOR I = 1 TO NP:
    PRINT  CR$(  FN R(GF(I)))GF(I);:
NEXT I:
PRINT :
& L(255):
POKE 34,2:
GOTO 6
```

List the global stiffness matrix.

```
158  HOME :
     & L(P%):
     PRINT  TAB( 9)"GLOBAL STIFFNESS MATRIX" TAB( 39)" [M]":
     POKE 34, PEEK (37):
     FOR I = 1 TO NP:
         INVERSE :
         PRINT "ROW "I TAB( 39)" ":
         NORMAL :
         FOR J = 1 TO BW:
             PRINT  CR$(  FN R(GS(I,J)))GS(I,J);:
         NEXT J:
         PRINT :
     NEXT I:
     & L(255):
     POKE 34,2:
     GOTO 6
```

List the displacement components.

```
160  K2 =  INT (K / 2 +  .001):
     ON 2 * K2 = K GOTO 18:
     & L(P%):
     INVERSE :
     PRINT "NODE    "DI$(1)"-DISPLACEMENT    "DI$(2)
        "-DISPLACEMENT" TAB( 39)" ":
     NORMAL :
     PRINT " "(I +  1) / 2 TAB( 8)UU(I) TAB( 25)UU(I +  1):
     & L(255):
     RETURN
```

Retrieve data files (RE=5,6).

```
162  RE = 5:
     GOSUB 180:
     GOSUB 172:
     ON SI%(RE) = 0 GOTO 168:
     ONERR  GOTO 200

164  EF = 0:
     GOSUB 184:
     & R(EN%,N$):
     ON EF GOTO 164:
     GOSUB 190:
     RE = 6:
     GOSUB 172:
     ON SI%(RE) = 0 GOTO 168

166  EF = 0:
     GOSUB 184:
     & R(NC,N$):
     ON EF GOTO 166:
     GOTO 190
```

(168-170) Error messages.

```
168  FLASH :
     ON  PEEK (222) = 77 GOTO 170:
     PRINT "[G][G][M]DATA NOT DEFINED":
     NORMAL :
     PRINT "EITHER FILE":
     INVERSE :
     PRINT NA$:
     NORMAL :
     PRINT "DOESN'T EXIST OR THE DATA IS NOT VALID":
     GOSUB 4:
     BC% = 2:
     RETURN

170  PRINT "[G]SORRY, OUT OF MEMORY":
     NORMAL :
     END
```

Read information about file referenced by record RE.

```
172  N1 =  FRE (0):
     PRINT D$:
     PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"READ"FI$",R"RE:
     INPUT SI%(RE),NA$,DE$,N1,D1$,N2,D2$:
     PRINT D$"CLOSE"FI$:
     GOTO 186
```

Request filename for data to be saved.

```
174  ON  PEEK (2163) > = 2 GOTO 188:
     PRINT "[M]*** ";:
     INVERSE :
     PRINT "SAVE";:
     NORMAL :
     PRINT " DATA TO DISKETTE ***":
     ON FR GOTO 186:
     ON Z GOTO 192:
     & I("[M]FILE NAME FOR[M]" +  DE$,NA$,NA$,"10",BC%,30):
     ON BC% < >0 GOTO 174:
     GOTO 186
```

Set active status and write information about data file RE.

```
176  SI%(RE) = 1:
     ON  PEEK (2163) = 3 GOTO 18:
     PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"WRITE"FI$",R"RE:
     PRINT SI%(RE):
     PRINT NA$:
     PRINT DE$:
     PRINT N1:
     PRINT D1$:
     PRINT N2:
     PRINT D2$:
     PRINT D$"CLOSE"FI$

178  PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2" "D2$"[M]
        WERE SAVED"CR$(FR = 0):
     ON FR = 0 GOTO 18:
     IF FR = 1 THEN
        PRINT " IN FILE: "NA$:
        RETURN
```

(180-182) Verify the existence of data file.

```
180  ONERR  GOTO 208
```

```
182  E2 = 0:
     PRINT D$:
     PRINT D$"CKFILE"FI$;DR$(DR):
     ON E2 GOTO 182:
     RETURN
```

(184-188) Request name of data file to be retrieved.

```
184  ON Z = 2 GOTO 188:
     PRINT "[M]*** ";:
     INVERSE :
     PRINT "LOADING";:
     NORMAL :
     PRINT " DATA FROM DISKETTE ***":
     ON FR GOTO 188:
     ON Z GOTO 192:
     & I("[M]FILE NAME FOR[M]" +  DE$,NA$,NA$,"11",BC%,30):
     ON BC% < >0 GOTO 184

186  ON  LEFT$ (NA$, LEN (KW$)) = KW$ GOTO 188:
     NA$ = KW$ +  "/" +  NA$

188  N$ = NA$ +  DR$(DR):
     RETURN
```

(190-192) Describe data file just loaded.

```
190  PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2" "D2$"[M]
        WERE LOADED":
     ON FR = 0 GOTO 18:
     PRINT " FROM FILE: "NA$:
     RETURN

192  PRINT " ";NA$:
     GOTO 188
```

Read the problem description.

```
194  PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"READ"FI$",R0":
     INPUT TE,KW$,PD$:
     PRINT D$"CLOSE"FI$:
     RETURN
```

Print the problem description.

```
196  ON  LEN (KW$) = 0 GOSUB 194:
     PRINT "[M]PROBLEM KEYWORD: "KW$"[M][M]
        PROBLEM DESCRIPTION:[M]"PD$:
     RETURN
```

Disable data files.

```
198  ON PEEK(2163) = 3 GOTO 18:
     PRINT D$"OPEN"FI$",L100"DR$(DR):
     FOR RE = R1 TO R2:
         PRINT D$"WRITE"FI$",R"RE:
         PRINT 0:
         SI%(RE) = 0:
     NEXT RE:
     PRINT D$"CLOSE"FI$:
     RETURN
```

(200-210) Error traps.

```
200  EF = 1:
     POKE 216,0:
     ER =  PEEK (222):
     ON ER = 255 GOTO 204:
     FLASH :
```

```
     ON ER = 77 GOTO 170:
     PRINT "[G][G][M]ERROR # [M]"ER:
     NORMAL :
     ER = ER * (ER >3 AND ER <11)
        +   NOT (ER >3 AND ER <11):
     ON ER = 1 OR ER = 5 GOTO 202:
     ON ER >3 AND ER <11 GOTO 206

202  GOSUB 172:
     RESUME

204  END

206  PRINT ER$(ER - 4):
     ON ER = 6 GOSUB 210:
     ON ER = 4 OR ER = 5 OR ER = 8 OR ER = 9 GOSUB 4:
     ON ER <10 GOTO 202:
     & I("UNLOCK?";"Y";A$,"12",0,1):
     ON A$ = "N" GOTO 202:
     PRINT D$"UNLOCK "NA$;DR$(DR):
     GOTO 202

208  E2 = 1:
     POKE 216,0:
     ON  PEEK (222) = 255 GOTO 204:
     ON  PEEK (222) = 77 GOTO 170:
     PRINT "[G][G][M]DATA DISKETTE IS NOT IN DRIVE":
     GOSUB 4:
     RESUME

210  & I("[M]CATALOG?";"N";A$,"13",0,1):
     ON A$ = "N" GOTO 18:
     PRINT D$"CATALOG"DR$(DR):
     RETURN
```

(1,212-228) Cold start initialization.

```
212  HOME :
     INVERSE :
     FOR I = 1 TO 4:
        PRINT  TAB( 2)" " TAB( 38)" ":
     NEXT I:
     NORMAL :
     VTAB 2:
     HTAB 3:
     PRINT " APPLIED FINITE ELEMENT ANALYSIS: ":
     HTAB 3:
     PRINT "     AN APPLE " CHR$ (221) CHR$ (219)
       " IMPLEMENTATION       ":
     VTAB 6:
     HTAB 12:
     PRINT "COPYRIGHT, 1985[M]"

214  INVERSE :
     HTAB 14:
     PRINT "SOLVE";:
     NORMAL :
     PRINT " (4.00)[M][M]ABSTRACT:[M][M]
        FINITE ELEMENT EQUATIONS ARE DEFINED FOREACH
        ELEMENT, ASSEMBLED INTO A GLOBAL[M]
        MATRIX EQUATION, MODIFIED BY BOUNDARY[M]
        CONDITIONS, AND SOLVED FOR NODAL[M]DISPLACEMENTS.":
     GOSUB 4
```

```
216  & C(DR$,ER$,CR$):
     DIM DR$(2),ER$(6),CR$(1):
     ER$(0) = "DISKETTE IS WRITE PROTECTED":
     DR$(1) = ",D1":
     DR$(2) = ",D2":
     DR =  PEEK (2048):
     ON DR = 2 GOTO 218:
     & B(200,20):
     PRINT "INSERT ";:
     FLASH :
     PRINT "DATA";:
     NORMAL :
     PRINT " DISKETTE INTO DRIVE":
     GOSUB 4

218  HOME :
     GOSUB 180:
     HOME :
     GOSUB 196:
     U$ = "----------------------------------------":
     PRINT U$:
     POKE 34,7:
     HOME :
     ER$(2) = "FILE NOT FOUND":
     ER$(3) = "VOLUME MISMATCH":
     ER$(4) = "I/O ERROR - DOOR/DISK INIT":
     ER$(5) = "DISK IS FULL":
     ER$(6) = "FILE IS LOCKED"

220  DIM SI%(25):
     RE = 5:
     GOSUB 172:
     NE = N1:
     DE = N2:
     DK = 2 * DE:
     RE = 6:
     GOSUB 172:
     NN = N1:
     NP = 2 * NN:
     DIM EN%(NE,DE),NC(NN,2),MP(NE,5),BC(NP,2):
     GOSUB 162:
     ON BC% = 2 GOTO 204:
     BW = 2 * EN%(0,0):
     DD = 3 * (TE = 1) +  4 * (TE = 2):
     DIM GF(NP),GS(NP,BW):
     ONERR  GOTO 200

222  DEF  FN R(X) =
          ( LEN ( STR$ (X)) +  PEEK (36) >39):
     CR$(0) = " ":
     CR$(1) =  CHR$ (13):
     DF$(1) = "X":
     DF$(2) = "Y":
     DF$(3) = "R":
     DF$(4) = "Z":
     DI$(1) = DF$(2 * TE - 1):
     DI$(2) = DF$(2 * TE):
     RE = 9:
     GOSUB 172

224  EF = 0:
     GOSUB 184:
     & R(MP,N$):
     ON EF GOTO 224:
```

```
      GOSUB 190:
      UN = (MP(0,1) * MP(0,2) * MP(0,5) * (MP(0,3) = 1
         OR TE = 2) * MP(0,4) = 1):
      RE = 11:
      GOSUB 172

226   EF = 0:
      GOSUB 184:
      & R(BC,N$):
      ON EF GOTO 226:
      GOSUB 190:
      P2 = 8 *  ATN (1):
      & C(PN$):
      DIM PN$(1):
      PN$(0) = "HELLO":
      PN$(1) = "POSTPROCESS"

228   DEF  FN P1(I) = I - 1 +  DE * (I = 1):
      DEF  FN P2(I) = I +  1 - DE * (I = DE):
      IF TE = 1 THEN
           HOME :
           & I("[M]ASSUME:[M]
              1. PLANE STRESS OR[M]
              2. PLANE STRAIN[M]" +  SR$,1;PS,"14",BC%,1,
              PS >0 AND PS <3):
           ON BC% GOTO 212,228,228
```

(2,230-236) Warm restart and menu.

```
230   FR = 0:
      POKE 216,0:
      TEXT :
      HOME :
      NM =  PEEK (2166):
      EO = 1:
      ON Z = 2 AND NM > = 6 GOTO 240:
      MO = 1:
      FR = (Z = 1 OR Z = 2)

232   PRINT "*** PROBLEM SOLUTION (4.05) ***[M][M]
         -- FORM & SOLVE ALL EQUATIONS --[M]
         1. DO COMPLETE PROBLEM SOLUTION[M]
         (MENU OPTIONS 2 - 5)[M][M]
         -- FORM SYSTEM EQUATIONS --[M]
         2. ASSEMBLE GLOBAL FORCE AND STIFFNESS

234   PRINT "    MATRICES[M]
         3. APPLY EXTERNAL FORCE BOUNDARY[M]   CONDITIONS[M]
         4. APPLY DISPLACEMENT BOUNDARY[M]   CONDITIONS[M][M]
         -- SOLVE SYSTEM EQUATIONS --[M]
         5. SOLVE FOR NODE DISPLACEMENTS[M][M]
         -- LIST RESULTS --[M]
         6. LIST NODE DISPLACEMENTS[M]"

236   PRINT "0. NONE OF THE ABOVE[M]":
      & I(SR$,NM;MO,"15",BC%,1,MO > = 0 AND MO <7):
      ON BC% >0 GOTO 228:
      ON MO = 0 GOTO 238:
      ON MO GOSUB 246,44,72,94,128,150:
      GOTO 230
```

(238-242) Exit menu.

```
238   HOME :
      PRINT "***          SOLVE: EXIT   (4.7 )         ***[M]
         1. PROCEED TO 'POSTPROCESS' PROGRAM[M]
```

```
       2. EXIT TO PROGRAM MENU[M]
       3. DON'T EXIT; REMAIN IN 'SOLVE'[M][M]0. STOP[M]":
    & I(SR$,1;EO,"16",BC%,1,EO > = 0 AND EO <4)

240 ON EO = 3 GOTO 230:
    ON BC% = 2 OR EO = 0 GOTO 204:
    POKE 2166,1:
    IF DR = 1 THEN
         & B(200,20):
        PRINT "[M]INSERT ";:
        FLASH :
        PRINT "PROGRAM";:
        NORMAL :
        PRINT " DISKETTE CONTAINING[M]'";:
        INVERSE :
        PRINT PN$(EO = 1);:
        NORMAL :
        PRINT "' INTO DRIVE":
        FR = 0:
        GOSUB 4:
        ON BC% < >0 GOTO 238

242 POKE 103,1:
    POKE 104,64:
    POKE 16384,0:
    PRINT D$"RUN "PN$(EO = 1)",D1"
```

Beep and delay subroutine.

```
244 & B(10,10):
    & B(0,0,20):
    RETURN
```

(146-250) Menu option 1: Complete options 2 through 5.

```
246 POKE 34,0:
    HOME :
    ON Z = 2 GOTO 248:
    PRINT "DO COMPLETE PROBLEM SOLUTION (4.1)[M][M]
       OPTIONS:[M][M]
       1. RUN W/O USER INTERACTION[M]    (USING ALL DEFAULTS)[M]
       2. RUN WITH USER INTERACTION[M]":
    & I(SR$,1;FR,"17",BC%,1,FR >0 AND FR <3):
    ON BC% < >0 GOTO 18:
    FR = (FR = 1):
    HOME

248 PRINT "DOING MENU OPTION 2:[M]
       ASSEMBLY OF GLOBAL FORCE AND STIFFNESS[M]MATRICES":
    GOSUB 244:
    GOSUB 44:
    HOME :
    PRINT "DOING MENU OPTION 3:[M]
       APPLICATION OF EXTERNAL FORCE BOUNDARY[M]CONDITIONS":
    GOSUB 244:
    GOSUB 72:
    HOME :
    PRINT "DOING MENU OPTION 4:[M]"

250 PRINT "APPLICATION OF DISPLACEMENT BOUNDARY[M]
       CONDITIONS":
    GOSUB 244:
    GOSUB 94:
    HOME :
```

```
          PRINT "DOING MENU OPTION 5:[M]
            SOLVING FOR NODE DISPLACEMENTS":
          GOSUB 244:
          GOSUB 128:
          ON FR = 0 GOTO 18:
          & B(50,20,3):
          RETURN

65535 REM [M][M]1FEB85[M]JRC/DCD

END-OF-LISTING
```

POSTPROCESS.ELASTICITY

(1,188-206) Cold start initialization.

```
1  TEXT :
   & D(T):
   CLEAR :
   P% = 10:
   & L(255):
   D$ =  CHR$ (4):
   SR$ = "... SELECT BY NUMBER <RETURN>":
   FI$ = "FILEINFO.TXT":
   POKE 2164,5:
   Z =  PEEK (2163):
   GOTO 188:
   REM POST(E)
```

(2,208-212) Warm restart.

```
2  GOTO 208
```

Wait for user input, clear screen, and continue.

```
4  ON FR = 1 GOSUB 220:
   VTAB (FR = 1) +  24 * (FR = 0):
   CALL  - 958:
   ON FR = 1 GOTO 10:
   & B(20,20):
   & I("... WHEN READY, PRESS <RETURN>";"";
     A$,"1",BC%):
   HOME :
   RETURN
```

(6-10) Decompose conjugate global equations using Gauss Elimination Method.

```
6  FOR I = 1 TO NN - 1:
       VTAB ( PEEK (37)):
       PRINT "  WORKING ON ROW #"I" OF "NN:
       C1 = CS(I,1):
       MJ = I +  BW / 2 - 1:
       MJ = MJ * (MJ < = NN) +  NN * (MJ >NN):
       N = 0:
       MK = BW / 2:
       IF NN - I +  1 <BW / 2 THEN
           MK = NN - I +  1

8      FOR J = I +  1 TO MJ:
           MK = MK - 1:
           N = N +  1:
           C2 = CS(I,N +  1) / C1:
           FOR K = 1 TO MK:
               CS(J,K) = CS(J,K) - C2 * CS(I,N +  K):
           NEXT K:
           FOR L = 1 TO DD:
               RH(J,L) = RH(J,L) - C2 * RH(I,L):
   NEXT L,J,I:
   VTAB ( PEEK (37)):
   CALL  - 868

10 RETURN
```

(12-16) Form material property (elasticity) matrix for element EL.

```
12 & M(D = (0)):
   IF PS = 1 AND TE = 1 THEN
       R = MP(EL,1) / (1 - MP(EL,2) ^ 2):
```

```
        D(1,2) = MP(EL,2) * R:
        D(3,3) = R * (1 - MP(EL,2)) / 2

14  IF (PS = 2 AND TE = 1) OR TE = 2 THEN
        R = MP(EL,1) * (1 - MP(EL,2)) / ((1 +  MP(EL,2))
          * (1 - 2 * MP(EL,2))):
        K = R * MP(EL,2) / (1 - MP(EL,2)):
        FOR I = 1 TO DD - 2:
            FOR J = I +  1 TO DD - 1:
                D(I,J) = K:
    NEXT J,I:
    D(DD,DD) = R * (1 - 2 * MP(EL,2))
       / (2 * (1 - MP(EL,2)))

16  FOR I = 1 TO DD - 1:
        D(I,I) = R:
    NEXT I:
    FOR I = 1 TO DD - 2:
        FOR J = I +  1 TO DD - 1:
            D(J,I) = D(I,J):
    NEXT J,I:
    RETURN
```

(18-24) Find element centroid, element area, initial thermal strain, and shape function gradient matrix for element EL.

```
18  RC = 0:
    ZC = 0:
    FOR I = 1 TO DE:
        ND(I) = EN%(EL,I):
        X(I) = NC(ND(I),1):
        Y(I) = NC(ND(I),2):
        NS(2 * I - 1) = 2 * ND(I) - 1:
        NS(2 * I) = 2 * ND(I):
        RC = RC +  X(I):
        ZC = ZC +  Y(I):
    NEXT I:
    RC = RC / DE:
    ZC = ZC / DE:
    A2 = X(2) * Y(3) +  X(3) * Y(1) +  X(1) * Y(2)
      - X(2) * Y(1) - X(3) * Y(2) - X(1) * Y(3):
    ON C% = 1 GOTO 10

20  DT = MP(EL,4) - MP(0,0):
    K = (1 +  MP(EL,2) * (TE = 1 AND PS = 2)) * MP(EL,5) * DT:
    FOR I = 1 TO DD - 1:
        E0(I,1) = K:
    NEXT I:
    E0(DD,1) = 0:
    & M(B = (0))

22  FOR J = 1 TO DE:
        I =  FN P1(J):
        K =  FN P2(J):
        B(1,2 * J - 1) = (Y(K) - Y(I)) / A2:
        B(2,2 * J) = (X(I) - X(K)) / A2:
        B(DD,2 * J) = B(1,2 * J - 1):
        B(DD,2 * J - 1) = B(2,2 * J):
        IF TE = 2 THEN
            B(DD - 1,2 * J - 1) = ((X(K) * Y(I) - X(I)
              * Y(K)) / RC +  Y(K) - Y(I) +  (X(I) - X(K))
              * ZC / RC) / A2

24  NEXT J:
```

DD-1

fix same as p324 l 38

DD-1

```
    RETURN
```

(26-50) Menu option 2: Find element strains and stresses.

```
26  POKE 34,0:
    HOME :
    PRINT "SOLVE EL. STRAINS & STRESSES  (5.2)[M][M]":
    POKE 34,2:
    RE = 17:
    & C(D,B,X,Y,ND,UU,U):
    GOSUB 158:
    GOSUB 150:
    ON SI%(RE) = 0 GOTO 146:
    DIM D(DD,DD),B(DD,DK),X(DE),Y(DE),ND(DE),UU(NP),
       U(DK,1):
    ONERR  GOTO 176

28  EF = 0:
    GOSUB 162:
    & R(UU,N$):
    ON EF GOTO 28:
    POKE 216,0:
    GOSUB 170:
    GOSUB 4:
    & C(E0,E1,E2,SI,SN,SS):
    DIM E0(DD,1),E1(DD,1),E2(DD,1),SI(DD,1),SN(NE,DD),
       SS(NE,DD)

30  PS = UU(0):
    HOME :
    PO = 0:
    ON FR GOTO 32:
    & I("DISPLAY STRAINS & STRESSES AS OBTAINED","N";
       A$,"2",BC%,1):
    ON BC% GOTO 26,26,30:
    PO = (A$ = "Y")

32  HOME :
    EL = 1:
    ON UN = 1 AND (SI%(11) = 0 OR SI%(12) = 0) GOSUB 12:
    PRINT "CALCULATING ELEMENT STRAINS & STRESSES"

34  & M(SN = (0)):
    & M(SS = (0)):
    C% = 0:
    FOR EL = 1 TO NE:
        VTAB 4:
        PRINT " FOR ELEMENT "EL" OF "NE:
        ON UN = 0 GOSUB 12:
        GOSUB 18:
        FOR I = 1 TO DE:
            U(2 * I - 1,1) = UU(2 * ND(I) - 1):
            U(2 * I,1) = UU(2 * ND(I)):
        NEXT I

36      & M(E1 = B.U):
        & M(E2 = E1 - E0):
        & M(SI = D.E2):
        FOR I = 1 TO DD:
            SN(EL,I) = E1(I,1):
            SS(EL,I) = SI(I,1):
        NEXT I

38      ON PO GOSUB 48:
    NEXT EL:
```

```
       & C(X,Y,ND,D,B,E0,E1,E2,SI):
       HOME :
       RE = 18:
       GOSUB 158:
       GOSUB 150:
       N1 = NE:
       N2 = DD:
       ONERR  GOTO 176

40     EF = 0:
       GOSUB 152:
       ON  PEEK (2163) = 3 GOTO 42:
       & S(SN,N$)

42     ON EF GOTO 40:
       GOSUB 154:
       RE = 19:
       GOSUB 150:
       N1 = NE:
       N2 = DD

44     EF = 0:
       GOSUB 152:
       ON  PEEK (2163) = 3 GOTO 46:
       & S(SS,N$)

46     ON EF GOTO 44:
       POKE 216,0:
       GOSUB 154:
       & C(SN,SS):
       POKE 2166,3:
       GOTO 4
```

(48-50) List element strains and stresses in the coordinate directions.

```
48     POKE 34,5:
       VTAB 6:
       & L(P%):
       PRINT "STRAINS":
       INVERSE :
       FOR J = 1 TO DD:
           PRINT "   "DI$(J);:
       NEXT J:
       PRINT  TAB( 39)" [M]":
       NORMAL :
       FOR I = 1 TO DD:
           PRINT  SPC(  FN  R(SN(EL,I)))SN(EL,I);:
       NEXT I:
       PRINT

50     VTAB 12:
       PRINT "STRESSES":
       INVERSE :
       FOR J = 1 TO DD:
           PRINT "   "DI$(J);:
       NEXT J:
       PRINT  TAB( 39)" [M]":
       NORMAL :
       FOR I = 1 TO DD:
           PRINT  SPC(  FN  R(SS(EL,I)))SS(EL,I);:
       NEXT I:
       PRINT :
       & L(255):
       GOSUB 4:
       POKE 34,2:
```

```
    RETURN
```

(52-68) Menu options 3 and 5: Find element and nodal principal stresses.

```
52  & C(T$):
    DIM T$(1):
    T$(0) = " ELEMENT ":
    T$(1) = " NODE ":
    POKE 34,0:
    HOME :
    PRINT "SOLVE"T$(MO = 5)"PRIN. STRESSES (5."NM")[M][M]":
    POKE 34,2:
    RE = 19 * (MO = 3) +  21 * (MO = 5):
    N = NE * (MO = 3) +  NN * (MO = 5):
    & C(SS):
    GOSUB 158:
    GOSUB 150:
    ON SI%(RE) = 0 GOTO 146:
    DIM SS(N,DD):
    ONERR  GOTO 176

54  EF = 0:
    GOSUB 162:
    & R(SS,N$):
    ON EF GOTO 54:
    POKE 216,0:
    GOSUB 170:
    GOSUB 4

56  HOME :
    PO = 0:
    ON FR GOTO 58:
    & I("DISPLAY PRINCIPAL STRESSES?";"N";A$,"3",BC%,1):
    ON BC% GOTO 26,26,52:
    PO = (A$ = "Y")

58  HOME :
    N = NE * (MO = 3) +  NN * (MO = 5):
    & C(PS):
    DIM PS(N,6):
    FOR NO = 1 TO N:
        IF PO = 0 THEN
            VTAB 3:
            PRINT "SOLVING FOR"T$(MO = 5)NO" OF "N

60      SM = .5 * (SS(NO,1) +  SS(NO,2)):
        TM =  SQR ((.5 * (SS(NO,1) - SS(NO,2))) ^ 2
            +  SS(NO,DD) ^ 2):
        S1 = SM +  TM:
        S2 = SM - TM:
        AN = 90:
        IF  ABS (SS(NO,1) - SS(NO,2)) >.001 THEN
            AN = RD *  ATN (2 * SS(NO,DD) / (SS(NO,1)
                - SS(NO,2)))

62      PS(NO,1) = S1:
        PS(NO,2) = S2:
        PS(NO,3) = TM:
        PS(NO,4) = AN:
        PS(NO,5) = AN +  90:
            PS(NO,6) = AN +  45:
            ON PO GOSUB 68:
        NEXT NO:
        HOME :
        RE = 20 * (MO = 3) +  22 * (MO = 5):
```

$\sigma_{\theta\theta}$ σ_{rr} σ_{zz} σ_{rz}

S3 = SS(NO,DD-1)

```
     GOSUB 158:
     GOSUB 150:
     N1 = NE * (MO = 3) +  NN * (MO = 5):
     N2 = 6:
     ONERR  GOTO 176

64   EF = 0:
     GOSUB 152:
     ON  PEEK (2163) = 3 GOTO 66:
     & S(PS,N$)

66   ON EF GOTO 64:
     GOSUB 154:
     POKE 2166,MO +  1:
     GOTO 4

68   & L(P%):
     PRINT T$(MO = 5)NO:
     INVERSE :
     PRINT "[M][M]   MAX      MIN      SHEAR" TAB( 39)" [M]":
     NORMAL :
     PRINT "STRESS:":
     FOR J = 1 TO 3:
        PRINT  SPC(  FN R(PS(NO,J)))PS(NO,J);:
     NEXT J:
     PRINT "[M][M]"AN$":":
     FOR J = 1 TO 3:
        PRINT  SPC(  FN R(PS(NO,J +  3)))PS(NO,J +  3);:
     NEXT J:
     PRINT :
     & L(255):
     GOTO 4
```

(70-98) Menu option 4: Find nodal stresses.

```
70   POKE 34,0:
     HOME :
     PRINT "SOLVE FOR NODAL STRESSES  (5.4)[M][M]":
     POKE 34,2:
     RE = 19:
     & C(CE,RE,CS,RH,SS):
     GOSUB 158:
     GOSUB 150:
     ON SI%(RE) = 0 GOTO 146:
     & C(SS):
     DIM CE(DE,DE),RE(DE,DD),CS(NN,BW / 2),RH(NN,DD),
        SS(NE,DD):
     ONERR  GOTO 176

72   EF = 0:
     GOSUB 162:
     & R(SS,N$):
     ON EF GOTO 72:
     GOSUB 170:
     GOSUB 4

74   HOME :
     PO = 0:
     ON FR GOTO 76:
     & I("DISPLAY CONJUGATE MATRICES?";"N";A$,"4",BC%,1):
     ON BC% GOTO 26,26,70:
     PO = 2 * (A$ = "Y")

76   HOME :
     PRINT "ASSEMBLING CONJUGATE MATRICES":
     POKE 34,3:
```

```
        C% = 1:
        FOR EL = 1 TO NE:
            VTAB 4:
            PRINT " FOR ELEMENT "EL" OF "NE" ELEMENTS"

78          GOSUB 18:
            C1 = A2 / 12:
            C2 = A2 / 24:
            & M(CE = (C2)):
            K = A2/6:
            FOR I = 1 TO DE:
                CE(I,I) = C1:
                FOR J = 1 TO DD:
                    RE(I,J) = K*SS(EL,J):
            NEXT J,I:

80          ON PO <2 GOTO 84:
            & L(P%):
            PRINT "[M]ELEMENT CONJUGATE STIFFNESS":
            INVERSE :
            PRINT "COLUMNS:  ";:
            FOR J = 1 TO DE:
                PRINT EN%(EL,J)"    ";:
            NEXT J:
            PRINT  TAB( 39)" ":
            NORMAL :
            FOR I = 1 TO DE:
                PRINT "ROW "EN%(EL,I)":";:
                FOR J = 1 TO DE:
                    VTAB 8 +  (J >2) +  2 * (I - 1):
                    HTAB 10 +  16 * (J = 2 OR J = 4)

82                  PRINT CE(I,J):
            NEXT J,I:
            PRINT

84          FOR I = 1 TO DE:
                KI = EN%(EL,I):
                FOR L = 1 TO DD:
                    RH(KI,L) = RH(KI,L) +  RE(I,L):
                NEXT L:
                FOR J = 1 TO DE:
                    KJ = EN%(EL,J):
                    IF KJ > = KI THEN
                        K = KJ - KI +  1:
                        CS(KI,K) = CS(KI,K) +  CE(I,J):
                        IF PO >1 THEN
                             PRINT "UPDATED GLOBAL K("KI","
                                KJ") IS "CS(KI,K)

86          NEXT J,I:
            PRINT :
            & L(255):
            ON PO >1 GOSUB 4:
        NEXT EL:
        POKE 34,2:
        HOME :
        PRINT "CONJUGATE MATRICES ASSEMBLED":
        ON FR GOSUB 220:
        PO = 0:
        ON FR GOTO 88:
        & I("[M][M]DISPLAY NODE STRESSES?";"N";A$,"5",BC%,1):
        ON BC% GOTO 26,26,86:
        PO = (A$ = "Y")
```

```
88  HOME :
    PRINT "DECOMPOSING MATRICES...[M]":
    GOSUB 6:
    HOME :
    PRINT "[M]SOLVING FOR NODE STRESSES";:
    & C(ST):
    DIM ST(NN,DD)

90  FOR L = 1 TO DD:
        ST(NN,L) = RH(NN,L) / CS(NN,1):
        FOR K = 1 TO NN - 1:
            PRINT ".";:
            I = NN - K:
            MJ = (BW / 2) * (I + BW / 2 - 1 <= NN)
               + (NN - I + 1) * (I + BW / 2
               - 1 >NN):
            SU = 0:
            FOR J = 2 TO MJ:
                N = I + J - 1:
                SU = SU + CS(I,J) * ST(N,L):
            NEXT J:
            ST(I,L) = (RH(I,L) - SU) / CS(I,1):
        NEXT K:
    NEXT L:
    PRINT

92  & C(CS,CE,RH,RE,SS):
    ON PO GOSUB 98:
    POKE 34,2:
    HOME :
    RE = 21:
    GOSUB 158:
    GOSUB 150:
    SI%(RE) = 1:
    N1 = NN:
    N2 = DD:
    ONERR GOTO 176

94  EF = 0:
    GOSUB 152:
    ON PEEK (2163) = 3 GOTO 96:
    & S(ST,N$)

96  ON EF GOTO 94:
    GOSUB 154:
    POKE 34,0:
    POKE 2166,5:
    GOSUB 4:
    RETURN

98  HOME :
    & L(P%):
    INVERSE :
    PRINT "NODE      STRESSES:";:
    FOR J = 1 TO DD:
        PRINT "   "DI$(J);:
    NEXT J:
    PRINT TAB( 39)" ":
    NORMAL :
    POKE 34,3:
    FOR I = 1 TO NN:
        PRINT I:
        FOR J = 1 TO DD:
            VTAB PEEK (37) + (J = 3):
```

```
            HTAB 9 +  16 * (J = 2 OR J = 4):
            PRINT ST(I,J):
     NEXT J,I:
     & L(255):
     RETURN
```

(100-112) Menu option 8: List the calculated results.

```
100  FR = 0:
     POKE 34,0:
     HOME :
     PRINT "LIST RESULTS (5.8)[M][M]":
     POKE 34,2:
     ON L2 = 0 GOSUB 158:
     PRINT "LISTING OPTIONS:[M]":
     FOR I = 1 TO 8:
         PRINT I". "LO$(I):
     NEXT I:
     PRINT "[M]0. NONE OF THE ABOVE[M]":
     & I(SR$,0;LO,"6",BC%,1,LO > = 0 AND LO <9):
     L2 = 1:
     ON BC% < >0 OR LO = 0 GOTO 10

102  POKE 34,0:
     HOME :
     INVERSE :
     PRINT "LIST "LO$(LO)"[M]":
     NORMAL :
     POKE 34,2:
     RE = 16 +  LO:
     GOSUB 150:
     ON SI%(RE) = 0 GOTO 146:
     DIM MO(N1,N2):
     SO = LO * (LO <3) +  3 * (LO >2 AND LO <7)
        +  4 * (LO = 7) +  5 * (LO = 8):
     ONERR  GOTO 176

104  IF N2 = 1 THEN
         & C(MO):
         DIM MO(N1)

106  EF = 0:
     GOSUB 162:
     & R(MO,N$):
     ON EF GOTO 106:
     GOSUB 170:
     GOSUB 4:
     PRINT LO$(LO):
     RW = (LO = 1) +  (LO >4 AND LO <9) +  2
        * (LO >1 AND LO <5):
     I = (LO = 1 OR LO = 8) +  2 * (LO = 2 OR LO = 3 OR
        LO = 5 OR LO = 7) +  3 * (LO = 4 OR LO = 6):
     ON I GOSUB 108,110,112:
     & C(MO):
     FR = 0:
     GOSUB 4:
     GOTO 100

108  & L(P%):
     INVERSE :
     PRINT RW$(RW)"     "S$(SO)": ";:
     FOR I = 1 TO 2:
         PRINT DI$(I)"  ";:
     NEXT I:
```

```
        PRINT  TAB( 39)" ":
        NORMAL :
        POKE 34,3:
        FOR I = 2 TO N1 STEP 2:
            PRINT " "I / 2 TAB( 7)MO(I - 1) TAB( 25)MO(I):
        NEXT I:
        & L(255):
        RETURN

110     & L(P%):
        INVERSE :
        PRINT RW$(RW)"    "S$(SO)": ";:
        FOR I = 1 TO N2:
            PRINT DI$(I)"  ";:
        NEXT I:
        PRINT  TAB( 39)" ":
        NORMAL :
        POKE 34,3:
        FOR I = 1 TO N1:
            PRINT " "I:
            FOR J = 1 TO N2:
                VTAB  PEEK (37) +  (J = 3):
                HTAB 8 +  16 * (J = 2 OR J = 4):
                PRINT MO(I,J):
        NEXT J,I:
        & L(255):
        RETURN

112     & L(P%):
        INVERSE :
        PRINT RW$(RW) TAB( 10)"STRESS" TAB( 18)AN$:
        NORMAL :
        POKE 34,3:
        FOR I = 1 TO N1:
            PRINT " "I;:
            FOR J = 1 TO 3:
                PRINT  TAB( 10)MO(I,J) TAB( 26)MO(I,J +  3):
        NEXT J,I:
        & L(255):
        RETURN
```

(114-128) Menu option 7: Find nodal reactions.

```
114     POKE 34,0:
        HOME :
        PRINT "EVALUATE NODAL REACTIONS (5.7)[M][M]":
        POKE 34,2:
        RE = 13:
        GOSUB 158:
        GOSUB 150:
        ON SI%(RE) = 0 GOTO 146:
        NP = N1:
        BW = N2:
        & C(GS,UU,RH):
        DIM GS(NP,BW),UU(NP),RH(NP):
        ONERR  GOTO 176

116     EF = 0:
        GOSUB 162:
        & R(GS,N$):
        ON EF GOTO 116:
        GOSUB 170:
        RE = 17:
        GOSUB 150:
        ON SI%(RE) = 0 GOTO 146
```

```
118  EF = 0:
     GOSUB 162:
     & R(UU,N$):
     ON EF GOTO 118:
     GOSUB 170:
     GOSUB 4:
     W = 1:
     Z = 0:
     HOME :
     FOR I = 1 TO NP:
        VTAB 3:
        PRINT "EVALUATING ROW "I" OF "NP

120     SU = Z:
        K = I - W:
        FOR J = 2 TO BW:
           M = J + I - W:
           RW = (M < = NP) + 2 * (K >Z):
           ON RW = Z GOTO 122:
           I1 = I * (RW < >2) + K * (RW = 2):
           M1 = M * (RW < >2) + K * (RW = 2):
           SU = SU + GS(I1,J) * UU(M1):
           K = K - (RW = 2):
           IF RW = 3 THEN
              SU = SU + GS(K,J) * UU(K):
              K = K - W

122     NEXT J:
        RH(I) = SU + GS(I,W) * UU(I):
     NEXT I:
     HOME :
     ON FR GOTO 124:
     & I("LIST REACTION";"N";A$,"7",0,1):
     IF A$ = "Y" THEN
         & L(P%):
        INVERSE :
        PRINT "NODE" TAB( 9)DI$(1) TAB( 25)DI$(2)
          TAB( 39)" ":
        NORMAL :
        POKE 34,3:
        FOR I = 2 TO NP STEP 2:
           PRINT " "I / 2 TAB( 9)RH(I - 1) TAB( 25)RH(I):
     NEXT

124  & L(255):
     POKE 34,2:
     ON FR = 0 GOSUB 4:
     RE = 24:
     GOSUB 150:
     N1 = NP:
     N2 = 1:
     ONERR  GOTO 176

126  EF = 0:
     GOSUB 152:
     ON  PEEK (2163) = 3 GOTO 128:
     & S(RH,N$)

128  ON EF GOTO 126:
     GOSUB 154:
     & C(GS,UU,RH):
     POKE 2166,8:
     GOTO 4
```

(130-138) Menu option 6: Find coordinates of the displaced nodes.

```
130  POKE 34,0:
     HOME :
     PRINT "CALC. NEW NODAL COORDINATES  (5.6)[M][M]":
     POKE 34,2:
     RE = 17:
     GOSUB 158:
     GOSUB 150:
     ON SI%(RE) = 0 GOTO 146:
     & C(UU,DC):
     DIM UU(NP),DC(NN,2):
     ONERR  GOTO 176

132  EF = 0:
     GOSUB 162:
     & R(UU,N$):
     ON EF GOTO 132:
     GOSUB 170:
     GOSUB 4

134  FOR I = 1 TO NN:
         FOR J = 1 TO 2:
             DC(I,J) = NC(I,J) +  UU(2 * I +  J - 2):
     NEXT J,I:
     RE = 23:
     GOSUB 150:
     N1 = NN:
     N2 = 2

136  EF = 0:
     GOSUB 152:
     ON  PEEK (2163) = 3 GOTO 138:
     & S(DC,N$)

138  ON EF GOTO 136:
     GOSUB 154:
     POKE 34,0:
     POKE 2165,6:
     POKE 2166,7:
     GOSUB 4:
     & C(DC):
     RETURN
```

(140-144) Retrieve element nodes and nodal coordinates.

```
140  RE = 5:
     GOSUB 158:
     GOSUB 150:
     ON SI%(RE) = 0 GOTO 146:
     ONERR  GOTO 176

142  EF = 0:
     GOSUB 162:
     & R(EN%,N$):
     ON EF GOTO 142:
     GOSUB 170:
     RE = 6:
     GOSUB 150:
     ON SI%(RE) = 0 GOTO 146

144  EF = 0:
     GOSUB 162:
     & R(NC,N$):
     ON EF GOTO 144:
```

```
    POKE 216,0:
    GOSUB 170:
    RETURN
```

(146-148) Error messages.

```
146 FLASH :
    ON  PEEK (222) = 77 GOTO 148:
    PRINT "[G][G][M]DATA NOT DEFINED":
    NORMAL :
    PRINT "EITHER THE FILE":
    INVERSE :
    PRINT NA$:
    NORMAL :
    PRINT "DOESN'T EXIST OR THE DATA IS NOT VALID":
    GOSUB 4:
    BC% = 2:
    RETURN

148 PRINT "[G]SORRY, OUT OF MEMORY":
    NORMAL :
    END
```

Read information about file referenced by record RE.

```
150 N1 =  FRE (0):
    PRINT D$"OPEN"FI$",L100"DR$(DR):
    PRINT D$"READ"FI$",R"RE:
    INPUT SI%(RE),NA$,DE$,N1,D1$,N2,D2$:
    PRINT D$"CLOSE"FI$:
    RETURN
```

Request name for data file to be saved.

```
152 ON  PEEK (2163) > = 2 GOTO 166:
    PRINT "[M]*** ";:
    INVERSE :
    PRINT "SAVE";:
    NORMAL :
    PRINT " DATA TO DISKETTE ***":
    ON FR GOTO 166:
    ON Z GOTO 168:
    & I("[M]FILE NAME FOR[M]" +  DE$,NA$,NA$,"8",BC%,30):
    GOTO 164
```

(154-156) Set active flag, update record RE file, and print message.

```
154 SI%(RE) = 1:
    ON  PEEK (2163) = 3 GOTO 10:
    PRINT D$"OPEN"FI$",L100"DR$(DR):
    PRINT D$"WRITE"FI$",R"RE:
    PRINT SI%(RE):
    PRINT NA$:
    PRINT DE$:
    PRINT N1:
    PRINT D1$:
    PRINT N2:
    PRINT D2$:
    PRINT D$"CLOSE"FI$:
    PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2" "D2$"[M]
       WERE SAVED":
    ON FR = 0 GOTO 10

156 IF FR = 1 THEN
        PRINT " IN FILE: "NA$:
      RETURN
```

(158-160) Verify existence of data file.

```
158  ONERR  GOTO 184

160  E2 = 0:
     PRINT D$"CKFILE"FI$;DR$(DR):
     ON E2 GOTO 160:
     RETURN
```

(162-168) Request name for data file to be retrieved.

```
162  ON Z = 2 GOTO 166:
     PRINT "[M]*** ";:
     INVERSE :
     PRINT "LOADING";:
     NORMAL :
     PRINT " DATA FROM DISKETTE ***":
     ON FR GOTO 166:
     ON Z GOTO 168:
     & I("[M]FILE NAME FOR[M]" +  DE$,NA$,NA$,"9",BC%,30)

164  ON  LEFT$ (NA$, LEN (KW$)) = KW$ GOTO 166:
     NA$ = KW$ +  "/" +  NA$

166  N$ = NA$ +  DR$(DR):
     RETURN

168  PRINT " ";NA$:
     GOTO 166
```

Describe data just loaded.

```
170  PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2" "D2$"[M]
        WERE LOADED":
     ON FR = 0 GOTO 10:
     PRINT " FROM FILE: "NA$:
     RETURN
```

Read problem description.

```
172  PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"READ"FI$",R0":
     INPUT TE,KW$,PD$:
     PRINT D$"CLOSE"FI$:
     RETURN
```

Print problem description.

```
174  ON  LEN (KW$) = 0 GOSUB 172:
     PRINT "[M]PROBLEM KEYWORD: "KW$"[M][M]
        PROBLEM DESCRIPTION:[M]"PD$:
     RETURN
```

(176-186) Error traps.

```
176  EF = 1:
     POKE 216,0:
     ER =  PEEK (222):
     ON ER = 255 GOTO 180:
     FLASH :
     ON ER = 77 GOTO 148:
     PRINT "[G][G][M]ERROR # "ER"[M]":
     NORMAL :
     ER = ER * (ER >3 AND ER <11)
        +  NOT (ER >3 AND ER <11):
     ON ER = 1 OR ER = 5 GOTO 178:
     ON ER >3 AND ER <11 GOTO 182

178  GOSUB 150:
     RESUME
```

```
180  END

182  POKE 216,0:
     PRINT ER$(ER - 4):
     ON ER = 6 GOSUB 186:
     ON ER = 8 GOSUB 4:
     ON ER <10 GOTO 178:
     & I("UNLOCK?";"Y";A$,"10",0,1):
     ON A$ = "N" GOTO 178:
     PRINT D$"UNLOCK "NA$DR$(DR):
     GOTO 178

184  E2 = 1:
     POKE 216,0:
     ON  PEEK (222) = 255 GOTO 180:
     ON  PEEK (222) = 77 GOTO 148:
     PRINT "[G][G][M]DATA DISKETTE IS NOT IN DRIVE":
     GOSUB 4:
     RESUME

186  & I("[M]CATALOG?";"N";A$,"11",0,1):
     ON A$ = "N" GOTO 10:
     PRINT D$"CATALOG"DR$(DR):
     RETURN
```

(1,188-206) Cold start initialization.

```
188  HOME :
     INVERSE :
     FOR I = 1 TO 4:
        PRINT  TAB( 2)" " TAB( 38)" ":
     NEXT I:
     NORMAL :
     VTAB 2:
     HTAB 3:
     PRINT " APPLIED FINITE ELEMENT ANALYSIS: ":
     HTAB 3:
     PRINT "     AN APPLE " CHR$ (221) CHR$ (219);
       " IMPLEMENTATION     ":
     VTAB 6:
     HTAB 12:
     PRINT "COPYRIGHT, 1985[M]"

190  HTAB 11:
     INVERSE :
     PRINT "POSTPROCESS";:
     NORMAL :
     PRINT " (5.00)[M][M]ABSTRACT:[M]
        NODE DISPLACEMENTS DETERMINED IN 'SOLVE'ARE USED
        TO DEFINE ELEMENT STRAINS,[M]
        ELEMENT STRESSES, AND NODE STRESSES.[M]
        STRESSES ARE DEFINED IN COORDINATE AND"

192  PRINT "PRINCIPAL DIRECTIONS.[M][M]
        IN ADDITION NODE REACTIONS AND NODE NEW POSITIONS
        ARE DETERMINED.":
     GOSUB 4:
     & C(ER$):
     DIM ER$(6):
     ER$(0) = "DISKETTE IS WRITE PROTECTED":
     ER$(2) = "FILE NOT FOUND":
     ER$(3) = "VOLUME MISMATCH":
     ER$(5) = "DISK IS FULL"

194  ER$(4) = "I/O ERROR - DOOR/DISK INIT":
```

```
     ER$(6) = "FILE IS LOCKED":
     & C(DR$):
     DIM DR$(2):
     DR$(1) = ",D1":
     DR$(2) = ",D2":
     DR =  PEEK (2048):
     ON DR = 2 GOTO 196:
     & B(200,20):
     PRINT "INSERT ";:
     FLASH :

     PRINT "DATA";:
     NORMAL :
     PRINT " DISKETTE INTO DRIVE":
     GOSUB 4

196  DEF  FN R(X) = 2 +  (40 -  PEEK (36))
        * ( LEN ( STR$ (X)) +   PEEK (36) >39):
     HOME :
     GOSUB 158:
     HOME :
     GOSUB 174:
     FOR I = 1 TO 38:
         PRINT "-";:
     NEXT :
     PRINT :
     POKE 34,7:
     HOME :
     VTAB 8

198  DIM SI%(25):
     RE = 5:
     GOSUB 150:
     NE = N1:
     DE = N2:
     DK = 2 * DE:
     RE = 6:
     GOSUB 150:
     NN = N1:
     NP = 2 * NN:
     DIM EN%(NE,DE),NC(NN,2),MP(NE,5):
     GOSUB 140:
     BW = 2 * EN%(0,0):
     DD = 3 * (TE = 1) +  4 * (TE = 2):
     CR$(0) = " ":
     CR$(1) =  CHR$ (13)

200  DF$(1) = "X":
     DF$(2) = "Y":
     DF$(3) = "R":
     DF$(4) = "Z":
     DI$(1) = DF$(2 * TE - 1):
     DI$(2) = DF$(2 * TE):
     DI$(3) = "THETA":
     DI$(DD) = DI$(1) +  "-" +  DI$(2)

202  GOSUB 222:
     UN = (MP(0,1) * MP(0,2) * MP(0,5)
        * (MP(0,3) = 1 OR TE = 2) * MP(0,4) = 1):
     RD = 22.5 /  ATN (1):
     DEF  FN P1(I) = I - 1 +  DE * (I = 1):
     DEF  FN P2(I) = I +  1 - DE * (I = DE):
     AN$ = "ANGLE CCW FROM " +  DI$(1) +  " (DEG)"
```

DD-1

(2)

```
204  DIM MU$(7),LO$(9),RW$(2),S$(5):
     MU$(1) = "ELEMENT STRAINS & STRESSES":
     MU$(2) = "ELEMENT PRINCIPAL STRESSES":
     MU$(3) = "NODE COORDINATE STRESSES":
     MU$(4) = "NODE PRINCIPAL STRESSES":
     MU$(6) = "NODE REACTIONS":
     MU$(5) = "NODE NEW POSITIONS":
     RW$(1) = "NODE"

206  RW$(2) = "ELEMENT":
     LO$(1) = "NODE DISPLACEMENTS":
     LO$(2) = "ELEMENT STRAINS":
     LO$(3) = "ELEMENT COORDINATE STRESSES":
     FOR I = 4 TO 8:
        LO$(I) = MU$(I - 2):
     NEXT I:
     S$(1) = "DISPLACEMENTS":
     S$(2) = "STRAINS":
     S$(3) = "STRESSES":
     S$(4) = "COORDINATES":
     S$(5) = "FORCES"
```

(2,208-212) Warm restart and menu.

```
208  FR = 0:
     POKE 216,0:
     TEXT :
     HOME :
     NM = PEEK (2166):
     EO = 1:
     ON Z = 2 AND PEEK (2166) = 8 GOTO 216:
     FR = (Z = 1 OR Z = 2):
     MO = 2:
     L2 = 0:
     ON Z = 2 GOTO 226

210  PRINT " POST-PROCESSING (5.05)[M][M]
        -- SOLVE ALL EQUATIONS --[M]
        1. DO ALL CALCULATIONS (MENU OPTIONS      2 - 7 )[M][M]
        -- SOLVE FOR STRAINS AND STRESSES --":
     FOR I = 2 TO 5:
        PRINT I". "MU$(I - 1):
     NEXT I

212  PRINT "[M]
        -- OTHER CALCULATIONS --[M]
        6. "MU$(5)"[M]
        7. "MU$(6)"[M][M]
        -- LISTING --[M]
        8. LIST OUTPUT[M][M][M]
        0. NONE OF THESE[M]":
     & I(SR$,NM;MO,"12",BC%,1,MO > = 0 AND MO <9):
     ON BC% GOTO 1,214,214:
     L2 = 0:
     ON MO + 1 GOTO 214,226:
     ON MO - 1 GOSUB 26,52,70,52,130,114,100:
     GOTO 208
```

(214-218) Exit menu.

```
214  HOME :
     PRINT "  POSTPROCESS: EXIT (5.9)[M]
        1. PROCEED TO 'PLOT' PROGRAM[M]
        2. EXIT TO MAIN MENU[M]
        3. DON'T EXIT; REMAIN IN 'POSTPROCESS'[M][M]
        0. STOP[M]":
```

```
      & I(SR$,1;EO,"13",BC%,1,EO > = 0 AND EO <4):
      ON BC% = 1 OR EO = 3 GOTO 208:
      ON BC% = 2 OR EO = 0 GOTO 180

216   POKE 2166,1:
      & C(PN$):
      DIM PN$(1):
      PN$(0) = "HELLO":
      PN$(1) = "PLOT":
      IF DR = 1 THEN
          & B(200,20):
          PRINT "[M]INSERT ";:
          FLASH :
          PRINT "PROGRAM";:
          NORMAL :
          PRINT " DISKETTE CONTAINING[M]'";:
          INVERSE :
          PRINT PN$(EO = 1);:
          NORMAL :
          PRINT "' INTO DRIVE ";:
          FR = 0:
          GOSUB 4:
          ON BC% GOTO 214

218   POKE 103,1:
      POKE 104,64:
      POKE 16384,0:
      PRINT D$"RUN "PN$(EO = 1)",D1"
```

Beep and delay subroutine.

```
220   & B(10,10):
      & B(0,0,20):
      RETURN
```

(222-224) Retrieve material properties data.

```
222   RE = 9:
      GOSUB 158:
      GOSUB 150:
      ONERR  GOTO 176

224   EF = 0:
      GOSUB 162:
      & R(MP,NA$):
      ON EF GOTO 224:
      GOSUB 170:
      RETURN
```

(226-228) Menu option 1: Do options 2-7.

```
226   POKE 34,0:
      HOME :
      FR = 1:
      MO = 2:
      ON Z = 2 GOTO 228:
      & I(SR$,1;FR,"14",BC%,1,FR >0 AND FR <3):
      ON BC% < >0 GOTO 208:
      FR = (FR = 1)

228   POKE 34,0:
      HOME :
      PRINT "DOING MENU OPTION "MO":[M][M]  CALCULATING "
         MU$(MO - 1):
      GOSUB 220:
      ON MO - 1 GOSUB 26,52,70,52,130,114:
```

```
        ON BC% >0 GOTO 208:
        MO = MO +  1:
        ON MO <8 GOTO 228:
        ON FR = 0 GOTO 208:
        FR = 0:
        & B(50,20,3):
        GOTO 208

65535 REM [M][M]1FEB85[M]JRC/DCD

END-OF-LISTING
```

PLOT.ELASTICITY

(1,284-312) Cold start initialization.

```
1  CLEAR :
   TEXT :
   & D(T):
   P% = 10:
   & L(255):
   D$ =  CHR$ (4):
   WR$ = "... WHEN READY, PRESS <RETURN>":
   SR$ = "... SELECT BY NUMBER <RETURN>":
   U$ = "--------------------------------------":
   M$ = "USE <I><J><K><M>TO MOVE ":
   S$ = "<S>TO SELECT ":
   GOTO 284:
   REM PLOT.E
```

(2,314-316) Warm restart and menu.

```
2  GOTO 314
```

Wait for user response, clear screen, and continue.

```
4  PRINT :
   & B(20,20):
   VTAB 24:
   CALL  - 868:
   & I(WR$;"";A$,"1",BC%):
   HOME :
   RETURN
```

Get cursor movement keystroke; scale if the control key is used.

```
6  GET G$:
         & B(150,4):
   A% =  ASC (G$):
   C% = 1 +  6 * (A% < = 0 OR A% > = 27):
   A% = A% +  64 * (A% >0 AND A% <27):
   RETURN
```

Erase background for label L$.

```
8  HCOLOR= 0:
   X = XP +  3.5 *  LEN (L$) +  XA * (3.5
      *  LEN (L$) +  5):
   Y = X - 7 *  LEN (L$):
   XP = XP - (X - 276) * (X >276)
      + ( ABS (Y) +  3) * (Y < = 2):
   FOR J1 = 1 TO 9:
       HPLOT XP - 3.5 *  LEN (L$) - 1 +  XA
         * (3.5 *  LEN (L$) +  5),YP - 4.5 +  J1
         - YA * 5 TO XP +  3.5 *  LEN (L$) +
         +  XA * (3.5 *  LEN (L$) +  5),YP - 4.5
         +  J1 - YA * 5:
   NEXT J1:
   HCOLOR= 3
```

Draw label L$ on graphics screen.

```
10 X = XP +  3.5 *  LEN (L$) +  XA * (3.5
      *  LEN (L$) +  5):
   Y = X - 7 *  LEN (L$):
   XP = XP - (X - 276) * (X >276)
      + ( ABS (Y) +  3) * (Y < = 2):
   FOR J1 = 1 TO  LEN (L$):
```

```
XDRAW  ASC ( MID$ (L$,J1,1)) - 31 AT XP
  - 3.5 *  LEN (L$) +  7 * (J1 - 1)
  +  XA * (3.5 *  LEN (L$) +  5),YP +  3.
  5 - YA * 5:
NEXT J1:
RETURN
```

Draw a cursor symbol and point to an element or node.

```
12 XDRAW 66 AT XL(N),YL(N):
   GET G$:
   XDRAW 66 AT XL(N),YL(N):
   & B(150,4):
   A =  ASC (G$):
   N = N +  FN S(NO):
   ON A < >83 AND A < >81 GOTO 12:
   L$ = LA$(LA) +  STR$ ( ABS (I%(N))):
   XP = XL(N):
   YP = YL(N):
   ON A = 83 AND A%(N) = 0 GOSUB 10:
   ON A = 83 GOSUB 48:
   ON A < >81 GOTO 12:
   A = 0:
   RETURN
```

Draw ' + ' symbol at ends and plot the contour line within the element.

```
14 LO = 0:
   ON ZT >0 GOSUB 26:
   ON LO < >0 GOTO 24:
   SCALE= 2:
   FOR J = 1 TO 2:
      DRAW 64 AT EX(J),EY(J):
   NEXT J:
   SCALE= 1:
   GOSUB 34:
   ON PF >0 GOTO 24:
   CV(CL,0) = CL:
   CV(CL,1) = EX(1):
   CV(CL,2) = EY(1):
   PF = 1:
   RETURN
```

(16-18) Draw a symbol on contour line, use to point, and then label the contour.

```
16 XDRAW 66 AT CV(N,1),CV(N,2):
   GET G$:
   XDRAW 66 AT CV(N,1),CV(N,2):
   & B(150,4):
   A =  ASC (G$):
   N = N +  FN C(P):
   ON A < >83 AND A < >81 GOTO 16:
   CL = CV(N,0):
   CV = V0 +  CL * DV:
   L$(1) =  CHR$ (CL +  65):
   L$(2) =  STR$ (CV):
   L$ = L$(CO):
   XP = CV(N,1):
   YP = CV(N,2)

18 I%(N) = I%(N) * (A < >83 OR (A = 83 AND A%(N)))
      +  (A = 83 AND A%(N) = 0):
   ON A = 83 AND A%(N) = 0 GOSUB 10:
   ON A = 83 GOSUB 48:
   ON A < >81 GOTO 16:
   A = 0:
   RETURN
```

Locate the contour line ends on two sides of an element.

```
20  P = P + 1:
    FR = (CV - S1) / (S2 - S1 +  (S2 = S1)):
    FOR J = 1 TO 2:
        CL(P,J) = NC(N1,J) +  FR * (NC(N2,J) - NC(N1,J)):
    NEXT J:
    ON P = 1 GOTO 24:
    FOR J = 1 TO 2:
        EX(J) = CL(J +  (P = 3),1):
        EY(J) = CL(J +  (P = 3),2):
    NEXT J:
    GOTO 14
```

Locate the contour line ends when two nodes of the element have the desired contour value.

```
22  EX(1) = NC(N1,1):
    EY(1) = NC(N1,2):
    EX(2) = NC(N2,1):
    EY(2) = NC(N2,2):
    GOTO 14

24  RETURN
```

(26-32) Clip lines which extend outside the plot bounds.

```
26  C(1) = 1:
    C(2) = 1:
    FOR EN = 1 TO 2:
        D%(EN,1) = (EX(EN) <SL):
        D%(EN,2) = (EX(EN) >SR):
        D%(EN,3) = (EY(EN) <ST):
        D%(EN,4) = (EY(EN) >SB):
    NEXT EN:
    LO =  FN AO(I);
    ON LO = 1 GOTO 24

28  FOR EN = 1 TO 2:
      IF D%(EN,1) = 1 OR D%(EN,2) = 1 THEN
          TV = SL * D%(EN,1) +  SR * D%(EN,2):
          SP = (EY(2) - EY(1)) / (EX(2) - EX(1)):
          EY(EN) = EY(1) +  (TV - EX(1)) * SP:
          EX(EN) = TV:
          D%(EN,1) = 0:
          D%(EN,2) = 0:
          D%(EN,3) = (EY(EN) <ST):
          D%(EN,4) = (EY(EN) >SB):
          LO =  FN AO(I):
          ON LO = 1 GOTO 32:
          C(EN) = 2

30    IF D%(EN,3) = 1 OR D%(EN,4) = 1 THEN
          TV = ST * D%(EN,3) +  SB * D%(EN,4):
          SP = (EX(2) - EX(1)) / (EY(2) - EY(1)):
          EX(EN) = EX(1) +  (TV - EY(1)) * SP:
          EY(EN) = TV:
          D%(EN,1) = (EX(EN) <SL):
          D%(EN,2) = (EX(EN) >SR):
          D%(EN,3) = 0:
          D%(EN,4) = 0:
          LO =  FN AO(I):
          C(EN) = 2

32    EN = EN +  LO:
    NEXT EN:
    RETURN
```

Plot a solid line from point 1 to point 2.

```
34  HPLOT EX(1),EY(1) TO EX(2),EY(2):
    RETURN
```

Plot a dashed line from point 1 to point 2.

```
36  X1 = EX(1):
    X2 = EX(2):
    Y1 = EY(1):
    Y2 = EY(2):
    L =  FN LE(I):
    ND = 2 *  INT (L / 6 +  .5) +  1:
    SX = (X2 - X1) / ND:
    SY = (Y2 - Y1) / ND:
    FOR J = 1 TO ND STEP 2:
        HPLOT X1 +  (J - .8) * SX,Y1 +  (J - .8) * SY
          TO X1 +  (J - .2) * SX,Y1 +  (J - .2) * SY:
    NEXT J:
    RETURN
```

Find the centroid of the portion of an element which lies within the plot bounds.

```
38  C = 0:
    XC = 0:
    YC = 0:
    FOR LI = 1 TO DE:
        I1 = ND%(EL,LI):
        I2 = ND%(EL, FN P2(LI)):
        EX(1) = NC(I1,1):
        EY(1) = NC(I1,2):
        EX(2) = NC(I2,1):
        EY(2) = NC(I2,2):
        GOSUB 26:
        IF LO = 0 THEN
            FOR K = 1 TO 2:
              C = C +  C(K):
              XC = XC +  C(K) * EX(K):
              YC = YC +  C(K) * EY(K):
        NEXT K

40      LI = LI +  DE * (C >4):
    NEXT LI:
    ON C = 0 GOTO 24:
    XC = XC / C:
    YC = YC / C:
    LO =  FN W(A):
    RETURN
```

Determine the method of plotting according to the clipping needed.

```
42  ON (ZN = 0) * (ZT = 0) * LT GOTO 34,36:
    ON ZT >0 GOSUB 26:
    ON ((ZT = 0) +  (ZT >0) * (LO = 0))
      * LT GOSUB 34,36:
    RETURN
```

(44-46) Draw a symbol, move it, and select the limits for a plot.

```
44  XDRAW 67 AT XP,YP:
    GOSUB 6:
    XDRAW 67 AT XP,YP:
    XP = XP - (A% = 74) * C% * (XP >4 +  C%)
       +  (A% = 75) * C% * (XP <272 - C%):
    YP = YP - (A% = 73) * C% * (YP >7 +  C%)
       +  (A% = 77) * C% * (YP <185 - C%):
    & D(V):
    IF YP <155 THEN
        & D(G)
```

```
46  ON G$ < >"S" AND G$ < >"Q" AND G$ < >
       "<" AND G$ < >">" GOTO 44:
    XDRAW 65 AT XP,YP:
    RETURN
```

(48-52) Move a label and save coordinates of its new location.

```
48  A%(N) = 1:
    GOSUB 6:
    C = (A% < >81):
    D = (A% < >88):
    ON C * D GOSUB 10:
    XP = XP - (A% = 74) * C% * (XP >7 +  C%)
       +  (A% = 75) * C% * (XP <272 - C%):
    YP = YP - (A% = 73) * C% * (YP >7 +  C%)
       +  (A% = 77) * C% * (YP <185 - C%):
    I%(N) = I%(N) * D - I%(N) *  NOT D:
    ON C GOSUB 10:
    ON C GOTO 48:
    IF CF = 0 THEN
       XL(N) = XP:
       YL(N) = YP

50  A = 0:
    ON CF = 0 GOTO 24:
    CV(N,1) = XP:
    CV(N,2) = YP:
    RETURN
```

Set display to graphics with four text lines at the bottom.

```
52  VTAB 21:
    POKE 34,20:
    HOME :
    & D(G):
    RETURN
```

Set display to text with the top two lines protected and erase the rest.

```
54  POKE 34,2:
    HOME :
    & D(T):
    VTAB 4:
    RETURN
```

(56-70) Draw symbols for boundary conditions.

```
56  SF = MF * (MF <3) +  2.5 * (MF >2):
    FOR N = 1 TO NB:
       NO = BN%(N,1):
       XC = NC(NO,1):
       YC = NC(NO,2):
       ON  FN W(A) GOTO 78:
       D2 = 2 * NO:
       D1 = D2 - 1:
       ON BC(D1,1) = 0 AND BC(D2,1) = 0 GOTO 78:
       B = 0:
       A = 8

58      B = B +  1:
       I1 = IS(2 * B - 1):
       I2 = IS(2 * B):
       ON N >I2 GOTO 58:
       N1 = BN%(N +   FN M(B),1):
       A = 21:
       N2 = BN%(N +   FN M(B),1):
       X1 = X(N1,1):
       Y1 = X(N1,2):
```

```
        X2 = X(N2,1):
        Y2 = X(N2,2):
        A =  FN LE(I):
        NV(1) = (Y2 - Y1) / A:
        NV(2) = (X1 - X2) / A

60      ON BC(D1,1) < >1 AND
           BC(D2,1) < >1 GOTO 68:
        F1 = BC(D1,2) * (BC(D1,1) = 1):
        F2 = BC(D2,2) * (BC(D2,1) = 1):
        X1 =  FN PX(X(NO,1) - (F1 * NV(1) <0)
           * FM * F1):
        Y1 =  FN PY(X(NO,2) - (F2 * NV(2) <0)
           * FM * F2):
        ON X1 <2 OR X1 >278 OR Y1 >190
           OR Y1 <2 GOTO 64

62      X2 =  FN PX(X(NO,1) +  (F1 * NV(1) > = 0)
           * FM * F1):
        Y2 =  FN PY(X(NO,2) +  (F2 * NV(2) > = 0)
           * FM * F2):
        DX = .3 * (X2 - X1):
        DY = .3 * (Y2 - Y1):
        ON X2 >1 AND X2 <278 AND Y2 >1
           AND Y2 <190 GOTO 66

64      L =  SQR (F1 ^ 2 +  F2 ^ 2):
        X1 = NC(NO,1):
        Y1 = NC(NO,2):
        X2 = X1 +  (12 / L) *  ABS (F1) *  SGN (NV(1)):
        Y2 = Y1 - (12 / L) *  ABS (F2) *  SGN (NV(2)):
        HPLOT X1,Y1 TO X2,Y2:
        SCALE= 1:
        DRAW 67 AT X2,Y2:
        GOTO 68

66      HPLOT X1,Y1 TO X2,Y2 TO X2 - DX - .4 * DY,
           Y2 - DY +  .4 * DX TO X2 - .8 * DX,Y2 - .8
           * DY TO X2 - DX +  .4 * DY,
           Y2 - DY - .4 * DX TO X2,Y2

68      ON BC(D1,1) < >2 AND
           BC(D2,1) < >2 GOTO 78:
        ON BC(D1,1) < >2 OR
           BC(D2,1) < >2 GOTO 72:
        A = ( ABS (NV(1)) > ABS (NV(2))):
        X1 = NC(NO,1) -  SGN (NV(1)) * A:
        Y1 = NC(NO,2) +  SGN (NV(2)) *  NOT A

70      DX = 3 * SF * (X1 - NC(NO,1)):
        X2 = X1 - 2 * DX:
        DY = 3 * SF * (Y1 - NC(NO,2)):
        Y2 = Y1 - 2 * DY:
        HPLOT X1,Y1 TO X2 - DY,Y2 - DX TO X2 +  DY,Y2
           +  DX TO X1,Y1:
    NEXT N:
    RETURN
```

(72-78) Draw a displacement boundary condition symbol.

```
72  A = (NV(1) > - 1E - 5):
    A = (NV(2) <1E - 5):
    C2 = (BC(D2,1) = 2) * (A -  NOT A):
    RA = 2:
    ND = 8 * SF:
```

```
    DA = P2 / ND

74  FOR J =  - 1 TO 1 STEP 2:
      X1 = NC(NO,1) +  1.4 * SF * RA * (C1 +  J * C2):
      Y1 = NC(NO,2) +  1.4 * SF * RA * (C2 - J * C1):
      HPLOT X1,Y1:
      HPLOT X1 +  SF * RA,Y1:
      FOR K = 1 TO ND:
        HPLOT  TO X1 +  SF * RA *  COS (K * DA),
           Y1 - SF * RA *  SIN (K * DA):
    NEXT K,J

76  X1 = NC(NO,1):
    Y1 = NC(NO,2):
    HPLOT X1 +  2.8 * SF * RA * (C1 - C2),Y1 +  2.8
       * SF * RA * (C1 +  C2) TO X1 +  2.8 * SF * RA
       * (C1 +  C2),Y1 +  2.8 * SF * RA * (C2 - C1)

78  NEXT N:
    RETURN
```

(80-84) Draw a line respresenting direction and magnitude of principal stress at each element or node.

```
80  LF = MF * (MF <3) +  3 * (MF >2):
    A =  ABS (V0) * ( ABS (V0) > ABS (V1))
       +  ABS (V1) * ( ABS (V1) > ABS (V0)):
    FOR I = 1 TO NR:
      I0 = I - I * (I >NN):
      XC = NC(I0,1):
      YC = NC(I0,2):
      IF MO = 5 THEN
          XC = EC(I,1):
          YC = EC(I,2)

82    IF  FN W(A) = 0 THEN
        AN = RD * AN(I):
        DX = 9 * VA(I) * LF *  COS (AN) / A:
        DY = 9 * VA(I) * LF *  SIN (AN) / A:
        HPLOT XC +  DX,YC - DY TO XC - DX,YC +  DY:
        IF VA(I) <0 THEN
             DRAW 76 AT XC,YC

84  NEXT I:
    SCALE= 1:
    RETURN
```

(86-90) Shade elements to represent a magnitude of a stress or strain component and display a legend.

```
86  PRINT :
    FOR EL = 1 TO NE:
      N% = NE - EL +  1:
      ON ZT >0 GOSUB 168:
      VA = VA(EL):
      IF VA >= VO AND VA < = V1 THEN
         GOSUB 38:
        IF LO = 0 THEN
           KD =  INT ((NI - .0001) * (VA - V0)
              / (V1 - V0)):
           C% = CO(KD):
           O% = OP(KD):
           & F(C%,O%,XC,YC)

88    ON ZT >0 GOSUB 168:
```

```
      NEXT EL:
      GOSUB 162:
      POKE 34,2:
      HOME :
      PRINT "TYPE: ";:
      INVERSE :
      PRINT TS$(TS):
      NORMAL :
      PRINT "[M]COMPONENT: ";:
      INVERSE :
      PRINT F$:
      NORMAL :
      PRINT "[M]VALUE RANGES FOR SPARSE TO DENSE
         DOTS   OR DARK TO BRIGHT SCREEN:"

90    FOR I = 1 TO NI:
         PRINT "  "V0 +  (I - 1) * DV; TAB( 18)" TO "V0
            +  I * DV:
      NEXT I:
      GOSUB 166:
      & I("[M]TYPE CTRL-V TO VIEW,[M] ELSE <RETURN>";
         "";A$,"2"):
      GOSUB 166:
      RETURN
```

Draw a rectangle enclosing the plot.

```
92    HPLOT SL - 1,SB +  1 TO SL - 1,ST - 1 TO SR +  1,ST  - 1
            TO SR +  1,SB +  1 TO SL - 1,SB +  1:
      RETURN
```

(94-98) Draw lines for contours.

```
94    & C(CL,CV):
      DIM CL(4,2),CV(NI,2):
      FOR CL = 0 TO NI:
         CV(CL,0) = 99:
         PF = 0:
         N% = NI - CL +  1:
         GOSUB 168:
         CV = V0 + CL * DV:
         FOR EL = 1 TO NE:
            P = 0:
            FOR I = 1 TO DE:
               N1 = ND%(EL,I):
               N2 = ND%(EL, FN P2(I)):
               S1 = VA(N1):
               S2 = VA(N2):
               A% = ((S1 - CV) * (CV - S2) > = 0)
                  * (1 +  (S1 = S2))

96             ON A% GOSUB 20,22:
         NEXT I,EL:
         GOSUB 168:
      NEXT CL:
      & C(CL):
      & W((CV,0 TO NI,0),0,NI,N%,BR%,ER%):
      P = ER%:
      ON N% = 0 GOTO 24:
      GOSUB 52:
      & I("MARK HI AND LO VALUES?";"N";A$,"3",0,1):
      ON A$ = "N" GOTO 24:
      XA = 0:
      YA = .1:
      L$ = "LO"
```

```
98  XP = CV(0,1):
    YP = CV(0,2):
    ON  NOT (XP <SL OR XP >SR OR YP >SB
       OR YP <ST) GOSUB 10:
    L$ = "HI":
    XP = CV(P,1):
    YP = CV(P,2):
    ON  NOT (XP <SL OR XP >SR OR YP >SB
       OR YP <ST) GOSUB 10:
    RETURN
```

(100-104) Draw arrows to represent magnitude and direction of node displacements.

```
100  SCALE= 3:
     ROT= 0:
     FOR I = 1 TO NN:
         N2 = 2 * I:
         N1 = N2 - 1:
         X1 = NC(I,1):
         Y1 = NC(I,2):
         ON X1 <35 OR X1 >245 OR Y1 <15
            OR Y1 >175 GOTO 84:
         DRAW 64 AT X1,Y1:
         ON MO < >3 OR PD < >2 GOTO 84:
         X2 =  FN PX(X(I,1) +  2 * MD * ND(N1)):
         Y2 =  FN PY(X(I,2) +  2 * MD * ND(N2))

102      IF X2 <5 OR X2 >275 OR Y2 <5
            OR Y2 >187 THEN
             L =  FN LE(I):
             X2 = X1 + 12 * (X2 - X1) / L:
             Y2 = Y1 + 12 * (Y2 - Y1) / L:
             HPLOT X1,Y1 TO X2,Y2:
             SCALE= 1:
             DRAW 67 AT X2,Y2:
             SCALE= 3:
             GOTO 84

104      DX = .3 * (X2 - X1):
         DY = .3 * (Y2 - Y1):
         HPLOT X1,Y1 TO X2,Y2 TO X2 - DX - .4 * DY,Y2
            - DY +  .4 * DX TO X2 - .8 * DX,Y2 - .8
            * DY TO X2 - DX +  .4 * DY
         ,Y2 - DY - .4 * DX TO X2,Y2:
     NEXT I:
     SCALE= 1:
     RETURN
```

(106-108) Retrieve nodal equivalent boundary conditions (RE=11) and find the greatest force magnitude.

```
106  RE = 11:
     GOSUB 254:
     GOSUB 252:
     ON SI%(RE) = 0 GOTO 248:
     NP = N1:
     & C(BC):
     DIM BC(NP,2):
     ONERR  GOTO 272

108  EF = 0:
     GOSUB 258:
     & R(BC,N$):
     ON EF GOTO 108:
     GOSUB 264:
```

```
    FH = 0:
    FOR I = 1 TO NP:
        A = (BC(I,1) = 1 AND  ABS (BC(I,2)) >FH):
        FH = FH *  NOT (A) +   ABS (BC(I,2)) * A:
    NEXT I:
    FH = FH +  (FH = 0):
    GOTO 4
```

(110-112) Retrieve nodal displacements (RE=17) and determine the greatest displacement magnitude.

```
110 RE = 17:
    GOSUB 254:
    GOSUB 252:
    ON SI%(RE) = 0 GOTO 248:
    & C(ND):
    NP = N1:
    DIM ND(NP):
    ONERR  GOTO 272

112 EF = 0:
    GOSUB 258:
    & R(ND,N$):
    ON EF GOTO 112:
    & O(ND,1 TO NP,0,A):
    A = (ND(NP) +  ND(1) >0):
    DH = ABS( A *  ND(NP) +   NOT A  *  ND(1)):
    & R(ND,N$):
    GOSUB 264:
    GOTO 4
```

(114-116) Retrieve the strains and stresses.

```
114 GOSUB 254:
    GOSUB 252:
    ON SI%(RE) = 0 GOTO 248:
    NR = N1:
    N% = N2:
    & C(ST):
    DIM ST(NR,N%):
    ONERR  GOTO 272

116 EF = 0:
    GOSUB 258:
    & R(ST,N$):
    ON EF GOTO 116:
    GOSUB 264:
    FOR I = 1 TO NR:
        ST(I,0) = I:
    NEXT I:
    GOTO 4
```

(118-120) Display data range and request plot range.

```
118 NI = 10:
    DV = (V1 - V0) / NI:
    ON TP = 2 GOTO 24:
    PRINT "[M]SPECIFY PLOT RANGE AND INCREMENT.[M][M]
      DATA MINIMUM: "V0"[M]DATA MAXIMUM: "V1"[M][M]
      DEFAULT IS FULL RANGE WITH "NI"[M]INCREMENTS OF "DV:
    & I(" O.K.?";"Y";A$,"4",0,1):
    ON A$ < >"N" GOTO 24

120 & I("[M]MINIMUM:";V0;V0,"5",BC%,0,V0 <V1):
    ON BC% GOTO 118,24,120:
    & I("MAXIMUM:";V1;V1,"6",BC%,0,V1 >V0):
    ON BC% GOTO 120,24,120:
```

```
& I("INCREMENT:";DV;DV,"7",BC%,0,DV <(V1 - V0)
   AND DV > = (V1 - V0) / 10.0001):
NI = INT (1.0001 * (V1 - V0) / DV):
RETURN
```

(122-124) Plot the mesh.

```
122 FOR LI = 1 TO NL:
       FOR I = 1 TO 2:
          N% = LN%(LI,I):
          EX(I) = NC(N%,1):
          EY(I) = NC(N%,2):
          IF DG = 1 THEN
             L = SC * MD:
             EX(I) = EX(I) + L * ND(2 * N% - 1):
             EY(I) = EY(I) - L * ND(2 * N%)

124    NEXT I:
       GOSUB 42:
    NEXT LI:
    RETURN
```

(126-130) Plot the boundaries.

```
126 A = 21:
    FOR B = 1 TO NS:
       FOR N = IS(2 * B - 1) TO IS(2 * B):
          N1 = N + FN M(B):
          B1 = BN%(N,1):
          B2 = BN%(N1,1):
          EX(1) = NC(B1,1):
          EY(1) = NC(B1,2):
          EX(2) = NC(B2,1):
          EY(2) = NC(B2,2)

128       IF DG = 1 THEN
             L = SC * MD:
             D1 = 2 * B1:
             D2 = 2 * B2:
             EX(1) = EX(1) + L * ND(D1 - 1):
             EY(1) = EY(1) - L * ND(D1):
             EX(2) = EX(2) + L * ND(D2 - 1):
             EY(2) = EY(2) - L * ND(D2)

130       GOSUB 42:
    NEXT N,B:
    RETURN
```

Draw an array of cells representing the equal-sized zones of the plot.

```
132 HCOLOR= 3 * Z%(I,J):
    FOR I1 = 1 TO 3:
       HPLOT FN ZX(I) - 1, FN ZY(J) - 2 + I1
          TO FN ZX(I) + 1, FN ZY(J) - 2 + I1:
    NEXT I1:
    RETURN
```

(134-140) Specify a rectangular region for a zoom plot.

```
134 GOSUB 52:
    PRINT "DEFINE ";:
    INVERSE :
    PRINT "LOWER LEFT CORNER":
    NORMAL :
    PRINT M$"CURSOR[M]"S$"CORNER":
    POKE 34,23:
    GOSUB 4:
    ON BC% >0 GOTO 24:
```

```
         POKE 34,0:
         VTAB 2:
         & D(V)

136      XP = SL:
         YP = SB - 10:
         GOSUB 44:
         ON G$ = "<" OR G$ = ">" GOTO 24:
         X1 =  FN UX(XP):
         Y1 =  FN VY(YP):
         XC = XP:
         YC = YP

138      & D(G):
         VTAB 21:
         PRINT "DEFINE ";:
         INVERSE :
         PRINT "UPPER RIGHT CORNER":
         NORMAL :
         VTAB 24:
         CALL  - 868:
         VTAB 2:
         XP =  FN PX(UH):
         YP = ST:
         GOSUB 44:
         ON G$ = "<" GOTO 134:
         ON G$ = ">" GOTO 24:
         X2 =  FN UX(XP):
         Y2 =  FN VY(YP):
         XDRAW 65 AT XC,YC:
         XDRAW 65 AT XP,YP

140      GOSUB 158:
         HOME :
         GOSUB 160:
         GOSUB 158:
         ON G$ = "N" GOTO 134:
         UL = X1:
         UH = X2:
         VL = Y1:
         VH = Y2:
         GOSUB 236:
         SR =  FN PX(UH):
         ST =  FN PY(VH):
         RETURN
```

(142-152) Specify size of the equal-sized zone plots, select zones to be plotted, and plot.

```
142      GOSUB 52:
         & I("MAGNIFICATION? (INTEGER 2 TO 6)",2;MF,"8",
            BC%,1,MF >1 AND MF <7):
         ON BC% >0 GOTO 24:
         GOSUB 154:
         & D(V):
         GOSUB 160:
         A = (G$ = "N"):
         ON A GOSUB 154:
         ON A GOTO 142:
         A$ = "ZONE PLOTTING OPTIONS"

144      GOSUB 52:
         PRINT A$"[M]
            1. ALL ZONES
            3. SELECT ZONES[M]
            2. OMIT ZONES          0. PLOT NONE":
```

```
       & I(SR$;1;ZS,"9",BC%,1,ZS > = 0 AND ZS <4):
       ON BC% >0 GOSUB 154:
       ON BC% GOTO 142,24,144:
       & C(Z%):
       DIM Z%(MF,MF):
       IF ZS <2 THEN
            FOR I = 1 TO MF:
               FOR J = 1 TO MF:
                    Z%(I,J) = 1:
       NEXT J,I:
       GOTO 154

146    GOSUB 52:
       PRINT M$"DOT IN CELLSAT TOP RIGHT CORNER, "S$"ZONES ";:
       INVERSE :
       PRINT ZS$(ZS - 2);:
       NORMAL :
       PRINT " PLOTTING,"Q$"[M]ZONE ";:
       FOR I = 1 TO MF - 1

148         HPLOT 250,13 +  4 * (I - 1) TO 252 +  4
               * (MF - 1),13 +  4 * (I - 1):
           HPLOT 253 +  4 * (I - 1),10 TO 253 +  4
               * (I - 1),12 +  4 * (MF - 1):
       NEXT I:
       I = 1:
       J = 1

150    HCOLOR= 3 - 3 * Z%(I,J):
       VTAB 24:
       HTAB 6:
       PRINT MF * (J - 1) +  I" ";:
       VTAB 2:
       HPLOT  FN ZX(I), FN ZY(J):
       GET G$:
       A =  ASC (G$):
       HCOLOR= 3 * Z%(I,J):
       HPLOT  FN ZX(I), FN ZY(J):
       I = I +   FN WM(I) * (G$ = "J" OR G$ = "K"):
       J = J +   FN WM(J) * (G$ = "I" OR G$ = "M"):
       IF G$ = "S" THEN
           Z%(I,J) =  NOT Z%(I,J):
           GOSUB 132

152    ON G$ < >"Q" GOTO 150:
       ON ZS = 3 GOTO 24:
       FOR I = 1 TO MF:
           FOR J = 1 TO MF:
               Z%(I,J) =  NOT Z%(I,J):
       NEXT J,I:
       RETURN
```

(154-156) Use lines of symbols to depict the equal-sized zones; label with zone numbers.

```
154    ROT= 16:
       FOR I = 1 TO MF - 1:
           PB = SB - I * (SB - ST) / MF:
           FOR J = SL TO SR STEP 15:
               XDRAW 69 AT J,PB:
       NEXT J,I:
       ROT= 0:
       FOR J = 1 TO MF - 1:
           PL = SL +  J * (SR - SL) / MF:
           FOR I = SB TO ST STEP  - 15:
               XDRAW 69 AT PL,I:
```

```
     NEXT I,J
```

Enclose single zone with symbols.

```
156  ZN = 0:
     FOR I = 1 TO MF:
         PB = SB - (I - 1) * (SB - ST) / MF:
         PT = PB - (SB - ST) / MF:
         FOR J = 1 TO MF:
             PL = SL + (J - 1) * (SR - SL) / MF:
             PR = PL + (SR - SL) / MF:
             ZN = ZN + 1:
             L$ = STR$ (ZN):
             XP = PL:
             YP = PT:
             XA = 1.5:
             YA = - 1.5:
             GOSUB 10:
             XP = PR:
             YP = PB:
             XA = - 1.5:
             YA = 1.5:
             GOSUB 10:
     NEXT J,I:
     RETURN
```

Enclose single zone with symbols.

```
158  PL = FN PX(X1):
     PR = FN PX(X2):
     PB = FN PY(Y1):
     PT = FN PY(Y2):
     FOR I = PT + 5 TO PB STEP 15:
         XDRAW 69 AT PL,I:
         XDRAW 69 AT PR,I:
     NEXT I:
     ROT= 16:
     FOR I = PL + 5 TO PR STEP 15:
         XDRAW 69 AT I,PT:
         XDRAW 69 AT I,PB:
     NEXT I:
     ROT= 0:
     RETURN
```

User approves the zoom region.

```
160  L$ = "O.K.? (Y/N)":
     XP = 15:
     YP = 191:
     XA = 1:
     YA = 1:
     & D(V):
     GOSUB 10:
     GOSUB 6:
     GOSUB 10:
     ON A% < >78 AND A% < >89 GOTO 160:
     RETURN
```

(162-168) Graphics screen messages.

```
162  L$ = "DRAWING SINGLE ZONE":
     ON ZT = 1 GOTO 164:
     ON ZN = 0 GOTO 24:
     L$ = "DRAWING ZONE " + STR$ (ZN) + "..."

164  XP = 52:
     YP = 0:
     XA = 1:
```

```
     YA = - 1:
     GOSUB 10:
     L$ = "PLEASE BE PATIENT":
     XP = 72:
     YP = 191:
     YA = 1:
     GOTO 10

166  L$ = "TYPE CTRL-T FOR TEXT, ELSE <RETURN>":
     XP = 1:
     YP = 191:
     XA = 1:
     YA = 1:
     GOTO 10

168  L$ =  STR$ (N%):
     XP = 260:
     YP = 91:
     XA = 1:
     YA = 1:
     GOTO 10
```

(170-184) Plot entire mesh, zoom plots.

```
170  GOSUB 54:
     PRINT "[M]PLOTS:[M]  WHOLE BODY[M]  ZONES (ZOOM)[M][M]
        LABELS:[M]  ELEMENTS[M]  NODES[M]  CONTOURS[M]
        NONE[M]  (USE RE-DRAW TO ENHANCE LABEL CLARITY)[M]":
     GOSUB 4:
     ON BC% GOTO 24

172  ZN = 0:
     SC = S0:
     UL = XL:
     UH = XH:
     VL = YL:
     VH = YH:
     MF = 1:
     SR = RS:
     ST = TT:
     ON ZT = 1 GOSUB 236:
     ZT = 0:
     GOSUB 186:
     ON BC% = 2 GOTO 184:
     GOSUB 54:
     HOME :
     PRINT "  THE WHOLE-BODY PLOT IS COMPLETE.[M][M]
        ZOOM OPTION: USER SPECIFIED RECTANGLE(S)FOR
        INCREASED LOCAL RESOLUTION."

174  PRINT "MULTIPLE PLOTS CAN BE COMBINED TO PRO- DUCE
        AN ENLARGED PLOT WITH INCREASED   RESOLUTION.":
     GOSUB 4

176  GOSUB 52:
     HOME :
     & I("ZONE PLOT OPTIONS:[M]
        1. SINGLE ZONE       0. NONE[M]
        2. MULTIPLE EQUAL-SIZED ZONES",1;ZT,"10",BC%,1,ZT > = 0
     AND ZT <3):
     ON BC% GOTO 24,24,176

178  ON ZT = 0 GOTO 24:
     ON ZT GOSUB 134,142:
     ON BC% = 1 GOTO 176:
```

```
        ON ZT = 1 AND BC% = 2 GOTO 24:
        MF = MF * (ZT = 2) +  2 * (ZT = 1):
        ON ZT = 1 GOSUB 186:
        ON ZT = 1 AND BC% = 2 GOTO 24:
        ON ZT = 1 GOTO 184:
        SC = S0 * MF:
        FOR ZJ = 1 TO MF:
            FOR ZI = 1 TO MF:
                ON Z%(ZI,ZJ) = 0 GOTO 182

180             ZN = (ZJ - 1) * MF +  ZI:
                V% = 1 +  (ZN - 1) / MF:
                H% = ZN - MF * (V% - 1):
                VL = YL +  (V% - 1) * (SB - ST) / SC:
                VH = YL +  V% * (SB - ST) / SC:
                UL = XL +  (H% - 1) * (SR - SL) / SC:
                UH = XL +  H% * (SR - SL) / SC:
                GOSUB 186:
                ON BC% = 2 GOTO 182:
                GOSUB 52:
                & I("REPEAT LAST ZONE";"N";A$,"11",0,1):
                ON A$ = "Y" GOTO 180

182             ZI = ZI +  MF * (BC% = 2):
            NEXT ZI:
            ZJ = ZJ +  MF * (BC% =2):
        NEXT ZJ

184     GOSUB 52:
        ON ZT = 0 GOTO 24:
        & I("[M]MORE ZONE PLOTS?";"N";A$,"12",0,1):
        ON A$ = "Y" GOTO 170:
        RETURN
```

(186-230) Produce the plot - including shading, arrows, contours, labels, corner markers, display, and saving to diskette.

```
186     SZ = 1:
        GOSUB 232:
        GOSUB 162:
        DG = 0:
        LT = 1 +  (MO = 3 AND PD = 1 AND GL = 1)
           +  (MO = 6 AND CP = 2):
        A% = 1 +  (MO = 1 OR (MO = 3 AND PD = 1 AND GD = 1)
           OR MO = 4 OR MO = 5 OR (MO = 6 AND CP = 2)):
        ON A% GOSUB 126,122:
        ON (MO = 5 OR MO = 6) AND TP = 2 GOSUB 80:
        ON MO = 6 AND TP = 3 GOSUB 94

188     IF MO = 3 THEN
            MD = (XH - XL + YH - YL) / (40 * DH):
            IF PD = 1 THEN
                LT = GL:
                DG = 1:
                ON GD GOSUB 122,126

190     IF (MO = 4 OR MO = 5) AND TP = 1 THEN
             HCOLOR= 3:
            ON MF >1 GOSUB 92:
            GOSUB 86:
            HCOLOR= 0:
            ON MF >1 GOSUB 92:
            HCOLOR= 3

192     ON MO = 3 AND PD = 2 GOSUB 100:
```

```
    FH = FH * (MO = 2) +  (MO < >2):
    FM = (XH - XL +  YH - YL) / (20 * FH):
    ON MO = 2 GOSUB 56:
    ON  NOT ((MO = 4 OR MO = 5) AND TP = 1) GOSUB 162:
    GOSUB 52:
    & I("LABEL PLOT (ANY TYPE)?";"N";A$,"13",0,1):
    ON A$ < >"Y" GOTO 224:
    LA = 1

194 GOSUB 52:
    & I("LABEL " +  PV$(LA) +  "S?";"N";A$,"14",0,1):
    ON A$ < >"Y" GOTO 208:
    CF = 0:
    GOSUB 54:
    PRINT "LABEL OPTIONS:[M][M]
       1. ALL "PV$(LA)"S (DEFAULT POSITIONS)[M]
       2. SELECTED "PV$(LA)"S (USER POSITIONS)[M][M]
       0. NONE[M]":
    N = 1

196 & I(SR$,1;LN,"15",BC%,1,LN > = 0 AND LN <3):
    ON BC% >0 OR LN = 0 GOTO 210:
    N = NE * (LA = 1) +  NN * (LA = 2):
    & C(A%,I%,XL,YL):
    DIM A%(N),I%(N),XL(N),YL(N):
    ON LA GOSUB 364,368:
    N = 1:
    ON LN = 2 GOTO 202:
    ON NO = 0 GOTO 200:
    & D(V):
    XA = (LA = 2):
    YA = 0

198 FOR I = 1 TO NO:
       L$ = LA$(LA) +   STR$ (I%(I)):
       XP = XL(I):
       YP = YL(I):
       GOSUB 10:
       A%(I) = 1:
    NEXT I:
    GOTO 206

200 HOME :
    VTAB 12:
    INVERSE :
    PRINT "NO "PV$(LA)"S FALL WITHIN THIS ZONE.":
    NORMAL :
    GOSUB 4:
    GOTO 208

202 GOSUB 54:
    PRINT "INSTRUCTIONS:[M]USE ARROWS TO MOVE CURSOR.[M]
      USE "S$"THE LABEL TO BE[M] CHANGED.[M]
      USE <X>TO ADD/REMOVE A LABEL.[M]
      "M$"IN LARGE[M] INCREMENTS.[M]
      USE <CONTROL>WITH THESE LETTERS TO
      MOVE IN SMALL INCREMENTS."

204 PRINT "USE"Q$" LABEL EDIT.":
    & I(WR$;"";A$,"16"):
    & D(V):
    N = N * (N >0 AND N < = NO)
       +  (N <1 OR N >NO):
```

```
    XA = (LA = 2):
    YA = 0:
    GOSUB 12

206 GOSUB 52:
    & I("EDIT " + PV$(LA) + " LABELS?";"N";A$,"17",0,1):
    ON A$ = "Y" GOTO 202:
    HOME :
    & I("REDRAW " + PV$(LA) + " LABELS?";"N";A$,"18",0,1):
    ON A$ < >"Y" GOTO 208:
    & D(V):
    FOR I = 1 TO NO:
        L$ = LA$(LA) + STR$ (I%(I)):
        XP = XL(I):
        YP = YL(I):
        ON I%(I) >0 AND A%(I) GOSUB 8:
    NEXT I

208 & D(G):
    HOME :
    LA = LA + 1:
    & C(A%,I%,XL,YL):
    ON LA <3 GOTO 194:
    LA = 1

210 CF = 1:
    ON TP < >3 OR MO < >6 GOTO 218:
    GOSUB 52:
    & I("LABEL CONTOUR LINES?";"N";A$,"19",BC%,1):
    ON BC% GOTO 194,24,210:
    ON A$ < >"Y" GOTO 218:
    HOME :
    PRINT "OPTIONS:[M]
       1. USE LETTER       0. NONE[M]
       2. USE VALUE":
    & I(SR$;1;CO,"20",BC%,CO = 0 OR CO = 1 OR CO = 2)


212 ON BC% GOTO 210,218,210:
    HOME :
    PRINT "ARROWS MOVE CURSOR[M]
       <S>SELECTS CONTOUR; <X>ADDS/DELETES
       <I><J><K><M>MOVE LABEL;"Q$:
    & I(WR$;"";A$,"21"):
    & D(V):
    XA = 0:
    YA = 0:
    & C(A%,I%):
    DIM A%(P),I%(P):
    FOR I = 0 TO P:
        I%(I) = - 1:
    NEXT I:
    N = 0:
    GOSUB 16

214 IF CO = 1 THEN
        POKE 34,2:
        HOME :
        INVERSE :
        PRINT GC$(GC * (MO = 5)):
        NORMAL :
        PRINT "LETTER      VALUE":
        FOR I = 0 TO NI:
            PRINT CHR$ (I + 65) TAB( 12)V0 + I * DV:
```

```
          NEXT I:
          GOSUB 166:
          VTAB 21:
          & I("CTRL-V TO VIEW,[M]ELSE <RETURN>";"";A$,"22"):
          & D(V):
          GOSUB 166

216       GOSUB 52:
          & I("REDRAW CONTOUR LABELS?";"N";A$,"23",0,1):
          ON A$ < >"Y" GOTO 218:
          & D(V):
          XA = 0:
          YA = 0:
          FOR N = 0 TO P:
              CL = CV(N,0):
              CV = V0 + CL * DV:
              L$(1) = CHR$ (CL + 65):
              L$(2) = STR$ (CV):
              L$ = L$(CO):
              XP = CV(N,1):
              YP = CV(N,2):
              ON I%(N) >0 GOSUB 8:
          NEXT N:
          & C(A%,I%):
          & D(G)

218       GOSUB 52:
          & I("OTHER LABELS?";"N";A$,"24",BC%,1):
          ON BC% GOTO 194,224,218:
          ON A$ < >"Y" GOTO 224:
          HOME :
          & I("LABEL:";"";L$,"25",BC%,30):
          ON BC% GOTO 218,224,218

220       HOME :
          PRINT "LOCATE CENTER OF LABEL-[M]"M$"CURSOR[M]"S$"POINT":
          VTAB 2:
          XA = 0:
          YA = 0:
          XP = 140:
          YP = 12:
          GOSUB 44:
          XDRAW 65 AT XP,YP:
          GOSUB 10:
          GOSUB 52:
          & I("EDIT LABEL?";"N";A$,"26",0,1):
          ON A$ < >"Y" GOTO 224:
          & D(V):
          N = 1:
          GOSUB 48

222       GOSUB 52:
          & I("REDRAW LABEL?";"N";A$,"27",0,1):
          ON A$ < >"Y" GOTO 218:
          GOSUB 8:
          GOTO 218

224       IF ZT = 2 THEN
              & D(V):
              SCALE= 1:
              DRAW 64 AT SL,ST:
              DRAW 64 AT SL,SB:
              DRAW 64 AT SR,ST:
              DRAW 64 AT SR,SB:
```

```
        GOSUB 52:
        PRINT "ZONE "ZN" IS COMPLETE.[M]
           TYPE CTRL-V TO VIEW FULL SCREEN.":
        GOSUB 4

226  GOSUB 54:
     PRINT "PLOT DISPOSITION:[M][M]
        1. PROCEED WITH ZONE(S)[M]
        2. SAVE PLOT TO DISKETTE[M][M]
        0. END THIS PLOT[M]":
     & I(SR$,0;A,"28",BC%,1,A > = 0 AND A <3):
     ON BC% GOTO 218,226:
     BC% = 2 * (A = 0):
     ON A <2 GOTO 24

228  HOME :
     PRINT "SAVE PICTURE[M][M] INSERT DATA DISKETTE, IF NECESSARY,[M]
        INTO DRIVE "DR"[M]":
     GOSUB 4:
     ON BC% GOTO 226,226,228:
     A$ =  MID$ (MU$(MO),6,17):
     & I("FILE NAME?[M]('" +  KW$ +  "/' PREFIX AND[M]
        '.PIC' SUFFIX WILL BE ADDED)",A$,A$,"29",BC%,17)

230  A$ = KW$ +  "/" +  A$ +  ".PIC":
     PRINT D$"BSAVE "A$",A$2000,L$1FF8":
     PRINT "[M]PLOT SAVED IN FILE:":
     INVERSE :
     PRINT A$:
     NORMAL :
     GOSUB 4:
     GOTO 226
```

(232-234) Calculate coordinates for scaled plot.

```
232  HGR :
     & D(G):
     VTAB 21:
     POKE 34,20:
     HOME :
     PRINT "CALCULATING COORDINATES...":
     XA = SL - SC * UL:
     YA = SB +  SC * VL:
     & M(NC = X * (SC)):
     FOR I = 1 TO NN:
         NC(I,1) = XA +  NC(I,1):
         NC(I,2) = YA - NC(I,2):
     NEXT I

234  & M(EC = CC * (SC)):
     FOR I = 1 TO NE:
         EC(I,1) = XA +  EC(I,1):
         EC(I,2) = YA - EC(I,2):
     NEXT I:
     HCOLOR= 3:
     SCALE= 1:
     ROT= 0:
     HOME :
     & D(V):
     RETURN
```

Define scale for plot.

```
236  S0 = (SB - ST) / (VH - VL):
     SC = (SR - SL) / (UH - UL):
     SC = SC * (SC < = S0) +  S0 * (SC >S0):
```

```
238  RE = 5:
     GOSUB 254:
     GOSUB 252:
     ON SI%(RE) = 0 GOTO 248:
     ONERR  GOTO 272
     S0 = SC:
     RETURN
```

(238-246) Retrieve element node numbers (RE=5), nodal coordinates (RE=6), line endpoints, boundary nodes; find ranges of coordinates and calculate element centroids.

```
240  EF = 0:
     GOSUB 258:
     & R(ND%,N$):
     ON EF GOTO 240:
     GOSUB 264:
     RE = 6:
     GOSUB 252:
     ON SI%(RE) = 0 GOTO 248:
     & C(X,NC,CC,EC):
     DIM X(NN,2),NC(NN,2),CC(NE,2),EC(NE,2)

242  EF = 0:
     GOSUB 258:
     & R(X,N$):
     ON EF GOTO 242:
     GOSUB 264:
     FOR I = 1 TO NN:
         X(I,0) = I:
         NC(I,0) = I:
     NEXT I:
     & O(X,1 TO NN,1,A):
     XL = X(1,1):
     XH = X(NN,1):
     & O(X,1 TO NN,2,A):
     YL = X(1,2):
     YH = X(NN,2):
     & O(X,1 TO NN,0,A):
     RE = 7:
     GOSUB 252:
     ON SI%(RE) = 0 GOTO 248

244  EF = 0:
     GOSUB 258:
     & R(LN%,N$):
     ON EF GOTO 244:
     GOSUB 264:
     RE = 8:
     GOSUB 252:
     ON SI%(RE) = 0 GOTO 248

246  EF = 0:
     GOSUB 258:
     & R(BN%,N$):
     ON EF GOTO 246:
     GOSUB 264:
     PRINT "[M]CALCULATING ELEMENT CENTROIDS...":
     & M(CC = (0)):
     FOR I = 1 TO NE:
         CC(I,0) = I:
         EC(I,0) = I:
         FOR J = 1 TO DE:
             K = ND%(I,J):
             CC(I,1) = CC(I,1) +  X(K,1):
             CC(I,2) = CC(I,2) +  X(K,2):
```

```
      NEXT J,I:
      & M(CC = CC / (DE)):
      RETURN
```

(248-252) Error messages.

```
248  FLASH :
     ON  PEEK (222) = 77 GOTO 250:
     PRINT "[G][G][M]DATA NOT DEFINED":
     NORMAL :
     PRINT "EITHER THE FILE":
     INVERSE :
     PRINT NA$:
     NORMAL :
     PRINT "DOESN'T EXIST OR THE DATA IS NOT VALID":
     GOSUB 4:
     BC% = 2:
     RETURN

250  PRINT "[G][G]SORRY, OUT OF MEMORY":
     NORMAL :
     END
```

Read information about file referenced in record RE.

```
252  N =  FRE (0):
     PRINT D$:
     PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"READ"FI$",R"RE:
     INPUT SI%(RE),NA$,DE$,N1,D1$,N2,D2$:
     PRINT D$"CLOSE"FI$:
     GOTO 260
```

Check for existence of data files.

```
254  ONERR  GOTO 280

256  E2 = 0:
     PRINT D$:
     ON E2 GOTO 256:
     RETURN
```

(258-262) Request name of data file to be loaded.

```
258  ON Z = 2 GOTO 262:
     PRINT "[M]*** ";:
     INVERSE :
     PRINT "LOADING";:
     NORMAL :
     PRINT " DATA FROM DISKETTE ***[M][M]":
     ON Z GOTO 266:
     & I("FILE NAME FOR[M]" +  DE$,NA$,NA$,"30",BC%,30):
     ON BC% >0 GOTO 258

260  ON  LEFT$ (NA$, LEN (KW$)) = KW$ GOTO 262:
     NA$ = KW$ +  "/" +  NA$

262  N$ = NA$ +  DR$(DR):
     RETURN
```

Describe data loaded.

```
264  PRINT "[M]"DE$"[M]FOR "N1" "D1$" AND "N2" "D2$"[M]
        WERE LOADED":
     RETURN

266  PRINT " ";NA$:
     GOTO 262
```

Read problem description.

```
268  PRINT D$"OPEN"FI$",L100"DR$(DR):
     PRINT D$"READ"FI$",R0":
     INPUT TE,KW$,PD$:
     PRINT D$"CLOSE"FI$:
     RETURN
```

Print problem description.

```
270  ON  LEN (KW$) = 0 GOSUB 268:
     PRINT "[M]PROBLEM KEYWORD: "KW$"[M][M]
        PROBLEM DESCRIPTION:[M]"PD$:
     RETURN
```

(272-282) Trap errors.

```
272  EF = 1:
     POKE 216,0:
     ER =  PEEK (222):
     ON ER = 255 GOTO 276:
     FLASH :
     ON ER = 77 GOTO 250:
     PRINT "[G][G][M]ERROR # "ER"[M]":
     NORMAL :
     ER = ER * (ER >3 AND ER <11)
        +  NOT (ER >3 AND ER <11):
     ON ER = 1 OR ER = 5 GOTO 274:
     ON ER >3 AND ER <11 GOTO 278

274  ON MO = 7 GOSUB 362:
     ON MO < >7 GOSUB 252:
     RESUME

276  PRINT "SO LONG.":
     END

278  PRINT ER$(ER - 4):
     ON ER = 6 GOSUB 282:
     ON ER = 4 OR ER = 5 OR ER = 8 OR ER = 9 GOSUB 4:
     ON ER <10 GOTO 274:
     & I("UNLOCK?";"Y";A$,"31",0,1):
     ON A$ = "N" GOTO 24:
     PRINT D$"UNLOCK "NA$;DR$(DR):
     GOTO 274

280  E2 = 1:
     POKE 216,0:
     ON  PEEK (222) = 255 GOTO 276:
     ON PEEK (222) = 77 GOTO 250:
     PRINT "[G][G][M]DATA DISKETTE IS NOT IN DRIVE":
     GOSUB 4:
     RESUME

282  & I("[M]CATALOG?";"N";A$,"32",0,1):
     ON A$ = "N" GOTO 24:
     PRINT D$"CATALOG"DR$(DR):
     RETURN
```

(1,284-312) Cold start initialization.

```
284  TEXT :
     & D(T):
     Q$ = " <Q>TO QUIT":
     POKE 2164,6:
     FI$ = "FILEINFO.TXT":
     Z =  PEEK (2163)
```

```
286  HOME :
     INVERSE :
     FOR I = 1 TO 4:
         PRINT  TAB( 2)" " TAB( 38)" ":
     NEXT I:
     NORMAL :
     VTAB 2:
     HTAB 3:
     PRINT " APPLIED FINITE ELEMENT ANALYSIS: ":
     HTAB 3:
     PRINT "     AN APPLE " CHR$ (221) CHR$ (219)
       " IMPLEMENTATION     ":
     VTAB 6:
     HTAB 12:
     PRINT "COPYRIGHT, 1985[M]"

288  HTAB 14:
     INVERSE :
     PRINT "PLOT";:
     NORMAL :
     PRINT "  (6.00)[M][M][M]ABSTRACT:[M][M]
       PREPARE FINISHED PLOTS OF GRID, BOUNDARYCONDITIONS,
       OR RESULTS.  ANY PLOT MAY BELABELLED AND PRINTED.
       ZONE PLOTS MAY BESELECTED TO ENLARGE PARTS OF INTEREST.":
     GOSUB 4

290  IF ( NOT ( PEEK (6462) = 75 AND  PEEK (6463) = 75)) THEN
         PRINT D$"BLOAD SHAPES,A$193E,D1":
         PRINT D$"BLOAD HI-RES DUMP,A$1D00"

292  & C(DR$):
     DIM DR$(2):
     DR$(1) = ",D1":
     DR$(2) = ",D2":
     DR =  PEEK (2048):
     ON DR = 2 GOTO 294:
     & B(200,20):
     PRINT "INSERT ";:
     FLASH :
     PRINT "DATA";:
     NORMAL :
     PRINT " DISKETTE INTO DRIVE":
     GOSUB 4

294  HOME :
     GOSUB 254:
     GOSUB 270:
     PRINT U$:
     & C(ER$):
     DIM ER$(6):
     ER$(0) = "DISKETTE IS WRITE PROTECTED":
     ER$(2) = "FILE NOT FOUND":
     ER$(3) = "VOLUME MISMATCH":
     ER$(4) = "I/O ERROR - DOOR OPEN OR DISK NOT INIT":
     ER$(5) = "DISK IS FULL":
     ER$(6) = "FILE IS LOCKED"

296  POKE 34,7:
     DD = 3 +  (TE = 2):
     DIM DI$(DD),DF$(5),SI%(24),CO(9),OP(9),ZS$(1),
       MU$(7),PV$(2),TP$(3),TS$(2),PC$(3),LA$(2),L$(2):
     DEF  FN C(P) = (A = 21) * (1 - (P +  1) * (N = P))
```

```
    + (A = 8) * ( - 1 + (P + 1) * (N = 0)):
   DEF FN LE(I) = SQR ((X1 - X2) ^ 2 + (Y1 - Y2) ^ 2)

298 DEF FN AO(I) = ((D%(1,1) * D%(2,1) + D%(1,2)
      * D%(2,2) + D%(1,3) * D%(2,3) + D%(1,4)
      * D%(2,4)) >0):
   RESTORE :
   DATA 20,0,10,1,19,0,5,1,16,1,8,1,8,0,1,1,1,0,0,0:
   FOR K = 0 TO 9:
      READ CO(K),OP(K):
   NEXT K:
   LA$(1) = "E":
   RD = ATN (1) / 45:
   P2 = 8 * ATN (1):
   DF$(1) = "X"
```

see 298 & 305 same

```
300 DF$(2) = "Y":
   DF$(3) = "R":
   DF$(4) = "Z":
   DF$(5) = "THETA":
   DI$(1) = DF$(2 * TE - 1):
   DI$(2) = DF$(2 * TE):
   DI$(3) = DF$(5 * (TE = 2)):
   DI$(DD) = DI$(1) + "-" + DI$(2):
   ZS$(0) = "OMITTED FROM":
   ZS$(1) = "INCLUDED IN"

302 DEF FN S(NE) = (A = 21) * (1 - NE * (N = NE))
    + (A = 8) * ( - 1 + NE * (N = 1)):
   DEF FN M(B) = ((N <IS(2 * B)) - (IS(2 * B)
    - IS(2 * B - 1)) * (N = IS(2 * B))) * (A = 21)
    - ((N >IS(2 * B - 1)) - (IS(2 * B)
    - IS(2 * B - 1)) * (N = IS(2 * B - 1))) * (A = 8):
   DEF FN P2(I) = I + 1 - DE * (I = DE):
   DEF FN W(A) = (XC <SL OR XC >SR
    OR YC >SB OR YC <ST)

304 RE = 5:
   GOSUB 252:
   NE = N1:
   DE = N2:
   RE = 6:
   GOSUB 252:
   NN = N1:
   RE = 7:
   GOSUB 252:
   NL = N1:
   RE = 8:
   GOSUB 252:
   NB = N1:
   NP = 2 * NN:
   DIM ND%(NE,DE),LN%(NL,3),BN%(NB,2),
      XP(DE),YP(DE),NV(2):
   GOSUB 238:
   ON BC% = 2 GOTO 276:
   NS = BN%(0,0):
   & C(IS):
   DIM IS(2 * NS)

306 FOR I = 1 TO 2 * NS:
      IS(I) = BN%(I,0):
   NEXT I:
   SL = 35:
   SR = 245:
```

```
SB = 175:
ST = 15:
RS = SR:
TT = ST:
DEF  FN PX(UX) = SL +  SC * (UX - UL):
DEF  FN PY(VY) = SB - SC * (VY - VL):
DEF  FN UX(PX) = UL +  (PX - SL) / SC:
DEF  FN VY(PY) = VL - (PY - SB) / SC:
UL = XL:
UH = XH:
VL = YL:
VH = YH:
GOSUB 236

308  MU$(0) = "EXIT":
     MU$(1) = "PLOT GENERATED GRID":
     MU$(2) = "PLOT BOUNDARY CONDITIONS":
     MU$(3) = "PLOT NODE DISPLACEMENTS":
     MU$(4) = "PLOT ELEMENT STRAINS":
     MU$(5) = "PLOT ELEMENT STRESSES":
     MU$(6) = "PLOT NODE STRESSES":
     MU$(7) = "RETRIEVE PICTURE FROM DISK"

310  PV$(1) = "ELEMENT":
     PV$(2) = "NODE":
     TP$(1) = "SHADE ELEMENT TO SHOW MAGNITUDE":
     TP$(2) = "LINE FOR DIRECTION & MAGNITUDE":
     TP$(3) = "CONTOUR LINES":
     TS$(1) = "COORDINATE-DIRECTION":
     TS$(2) = "PRINCIPAL":
     PC$(1) = "MAXIMUM PRINCIPAL STRESS"

312  PC$(2) = "MINIMUM PRINCIPAL STRESS":
     PC$(3) = "MAXIMUM SHEAR IN " + DI$(DD) +  " PLANE":
     DEF  FN WM(I) =  - (A = 74 OR A = 77) * ((1 - MF)
        * (I = 1) +  (I < >1)) +  (A = 75 OR A = 73)
        * ((1 - MF) * (I = MF) +  (I < >MF)):
     DEF  FN ZX(I) = 251 +  4 * (I - 1):
     DEF  FN ZY(J) = 11 +  4 * (MF - 1) - 4 * (J - 1)
```

(2,314-316) Warm restart and menu.

```
314  POKE 216,0:
     NM =  PEEK (2166):
     TEXT :
     & D(T):
     HOME :
     PRINT "      PLOT  (6.05)[M][M]
        -- PLOT FORMULATION --[M]
        1. "MU$(1)"[M]
        2. "MU$(2)"[M][M]
        -- PLOT RESULTS --[M]
        3. "MU$(3)"[M]
        4. "MU$(4)"[M]
        5. "MU$(5)"[M]
        6. "MU$(6)"[M][M]
        -- OTHER --"

316  PRINT
        "7. "MU$(7)"[M][M]
        0. NONE OF THESE[M]":
     & C(ST,VA,ND):
     & I(SR$,NM;MO,"33",BC%,1,MO > = 0 AND MO <8):
     HOME :
     ON MO = 0 GOTO 318:
```

```
HE$ = MU$(MO) +  "  (6.".+   STR$ (MO) +  "0)":
PRINT "**" TAB( (39 -  LEN (HE$)) / 2
   + 2)HE$ TAB( 39)"**":
POKE 34,2:
ON MO GOTO 324,324,342,328,326,326,350
```

(318-322) Exit menu.

```
318 HOME :
    PRINT "       PLOT: EXIT  (6.8)[M][M]
      1. EXIT TO MAIN MENU[M]
      2. DON'T EXIT; REMAIN IN 'PLOT'[M][M]
      0. STOP":
    & I(SR$,0;EO,"34",BC%,1,EO > = 0 AND EO <3):
    ON EO = 0 OR BC% = 2 GOTO 276:
    ON EO = 2 GOTO 314

320 IF DR = 1 THEN
        & B(200,20):
       PRINT "INSERT ";:
       FLASH :
       PRINT "PROGRAM";:
       NORMAL :
       PRINT " DISKETTE CONTAINING[M]'";:
       INVERSE :
       PRINT "HELLO";:
       NORMAL :
       PRINT "' INTO DRIVE ":
       GOSUB 4:
       ON BC% = 1 OR BC% = 3 GOTO 318


322 PRINT D$"RUN HELLO,D1"
```

Menu options 1 and 2: Plot generated grid and plot boundary conditions.

```
324 ON MO = 2 GOSUB 106:
    ON BC% = 2 GOTO 314:
    GOSUB 170:
    & D(T):
    & C(BC):
    SI%(RE) = 0:
    GOTO 314
```

(326-348) Menu option 5: Plot element stresses or nodal stresses.

```
326 POKE 34,2:
    HOME :
    PRINT "STRESS OPTIONS:[M][M]
      1. "TS$(1)"[M]
      2. "TS$(2)"[M][M]
      0. NONE OF THESE[M]":
    & I(SR$,1;TS,"35",BC%,1,TS > = 0 AND TS <3):
    ON BC% < >0 GOTO 326:
    IF TS = 0 THEN
       & C(ST):
      SI%(RE) = 0:
      GOTO 314
```

(328-340) Menu option 4: Plot element strains.

```
328 POKE 34,2:
    HOME :
    TS = TS +  (TS = 0):
    PRINT "TYPE: ";:
    INVERSE :
    PRINT TS$(TS):
    NORMAL :
```

```
        POKE 34,4:
        PRINT "[M]COMPONENT OPTIONS:[M]":
        ON TS = 2 GOTO 332:
        FOR I = 1 TO DD - 1:
            PRINT I". "DI$(I)"-COMPONENT":
        NEXT I:
        PRINT DD". "DI$(DD)" SHEAR[M][M]
           0. NONE OF THESE[M]"

330     & I(SR$,1;PC,"36",BC%,1,PC > = 0 AND PC < = DD):
        ON BC% < >0 GOTO 326:
        ON PC = 0 AND MO < >4 GOTO 326:
        A$ = DI$(PC):
        ON PC < >0 GOTO 334:
        IF PC = 0 AND MO = 4 THEN
             & C(ST):
            GOTO 314

332     FOR I = 1 TO 3:
            PRINT I". "PC$(I):
        NEXT I:
        PRINT "[M]0. NONE OF THESE[M]":
        & I(SR$,1;PC,"37",BC%,1,PC > = 0 AND PC <4):
        ON BC% < >0 GOTO 328:
        ON PC = 0 GOTO 326:
        A$ = PC$(PC)

334     POKE 34,4:
        HOME :
        PRINT "COMPONENT: ";:
        INVERSE :
        PRINT A$:
        NORMAL :
        POKE 34,6:
        TP = ((MO = 4 OR MO = 5) AND TS = 1) + 3
           * (MO = 6 AND TS = 1):
        ON TP = 1 GOTO 340:
        ON TP = 3 GOTO 338:
        HOME :
        PRINT "PLOT TYPE OPTIONS:[M]":
        FOR I = 1 TO 2:
            PRINT I". "TP$(I + (MO = 6)):
        NEXT :
        PRINT "[M]0. NONE OF THESE[M]"

336     & I(SR$,1;TP,"38",BC%,1,TP > = 0 AND TP <3):
        ON BC% < >0 GOTO 334:
        ON TP = 0 GOTO 328:
        TP = TP + (MO = 6):
        ON TP < >3 GOTO 340

338     HOME :
        PRINT "CONTOUR LINE OPTIONS:[M][M]
           1. PLOT ONLY CONTOUR LINES[M]
           2. PLOT CONTOUR LINES WITH DASHED MESH[M]":
        & I(SR$,1;CP,"39",BC%,1,CP >0 AND CP <3):
        ON BC% GOTO 334

340     HOME :
        RE = 18 + TS * (MO < >4) + 2 * (MO = 6):
        GOSUB 114:
        ON BC% = 2 GOTO 314:
        & C(VA,AN):
        DIM VA(NR),AN(NR):
```

```
FOR I = 1 TO NR:
     VA(I) = ST(I,PC):
     AN(I) = ST(I,PC +  3 * (TS = 2)):
NEXT I:
& O(ST,1 TO NR,PC,A):
V0 = ST(1,PC):
V1 = ST(NR,PC):
& C(ST):
GOSUB 118:
GOSUB 170:
& D(T):
& C(XY,VA,AN):
GOTO 328
```

(342-348) Menu option 3: Plot displacement of nodes.

```
342  POKE 34,2:
     HOME :
     PRINT "[M]DISPLACEMENT OPTIONS:[M][M]
        1. DISPLACED GRID OVER ORIGINAL[M]
        2. DISPLACEMENT VECTORS[M][M]
        0. NONE OF THESE[M]":
     & I(SR$,1;PD,"39",BC%,1,PD > = 0 AND PD <3):
     HOME :
     IF BC% OR PD = 0 THEN
          & C(ND):
          GOTO 314

344  ON PD < >1 GOTO 348:
     PRINT "[M]MESH OPTIONS:[M][M]1. ENTIRE GRID[M]
        2. ONLY BOUNDARY LINES[M][M]
        0. NONE OF THESE[M]":
     & I(SR$,1;GD,"40",BC%,1,GD > = 0 AND GD <3):
     ON BC% OR GD = 0 GOTO 342:
     HOME :
     PRINT "[M]LINE TYPE OPTIONS:[M]"

346  PRINT "[M]
        1. ORIGINAL: DASHED;[M]    DISPLACED: SOLID[M]
        2. ORIGINAL: SOLID;[M]     DISPLACED: DASHED[M][M]
        0. NONE OF THESE[M]":
     & I(SR$,1;GL,"41",BC%,1,GL > = 0 AND GL <3):
     ON BC% < >0 OR GL = 0 GOTO 344

348  POKE 34,2:
     HOME :
     GOSUB 110:
     ON BC% = 2 GOTO 314:
     GOSUB 170:
     & D(T):
     GOTO 314
```

(350-362) Menu option 7: Retrieve picture from disk.

```
350  POKE 34,2:
     HOME :
     & C(PF$):
     DIM PF$(20):
     ON DR = 2 GOTO 352:
     & B(200,20):
     PRINT "[M][M]INSERT ";:
     FLASH :
     PRINT "PROGRAM";:
     NORMAL :
     PRINT " DISKETTE":
     GOSUB 4:
```

```
    ON BC% GOTO 314:
    ONERR  GOTO 272

352 PRINT D$"BLOAD N,A$1420,D1":
    ON DR = 2 GOTO 354:
    PRINT "[M][M]INSERT ";:
    FLASH :
    PRINT "DATA";:
    NORMAL :
    PRINT " DISKETTE":
    GOSUB 4:
    ON BC% GOTO 314

354 PRINT D$"CATALOG"DR$(DR):
    & N(PF$,1,P,N%,SF%):
    PRINT "[M]"SF%" SECTORS FREE[M][M]
       "N% - 1" PICTURE(S)[M][M]":
    PF$(0) = "0. NONE":
    ON N% = 1 GOTO 356:
    FOR I = 1 TO N% - 1:
        PRINT I"." TAB( 5)PF$(I):
    NEXT I

356 PRINT "[M]"PF$(0)"[M]":
    & I(SR$,0;A,"42",0,1,A > = 0 AND A <N%):
    ON BC% GOTO 350,314,350

358 EF = 0:
    ON A = 0 GOTO 314:
    PRINT D$"BLOAD"PF$(A);DR$(DR):
    ON EF GOTO 358:
    HOME :
    VTAB 21:
    INVERSE :
    PRINT PF$(A);:
    NORMAL :
    PRINT " RETRIEVED[M]CTRL-T FOR TEXT (MENU);
       CTRL-V TO VIEW":
    & D(G):
    VTAB 3

360 PRINT "DISPOSITION OPTIONS:[M][M]
       1. PRINT, IF EPSON WITH GRAPPLER[M]
       2. RETRIEVE ANOTHER[M][M]
       0. NONE OF THESE[M][M]":
    & I(SR$,0;A,"43",BC%,1,A > = 0 AND A <3):
    ON BC% GOTO 350,314,350:
    ON A = 0 GOTO 314:
    ON A = 2 GOTO 354

362 PR# 1:
    PRINT  CHR$ (9);"GE":
    PR# 0:
    & D(T):
    GOTO 354
```

(364-366) Sort to identify elements with centroids inside the plot boundaries.

```
364 PRINT "[M]SORTING ELEMENTS...":
    & W((EC,1 TO NE,1),SL,SR,N%,BR%,ER%):
    & W((EC,BR% TO ER%,2),ST,SB,N%,BR%,ER%):
    NO = N%:
    IF N% >0 THEN
         & O(EC,BR% TO ER%,0,A):
```

```
          FOR I = 1 TO N%:
              N = I +  BR% - 1:
              XL(I) = EC(N,1):
              YL(I) = EC(N,2):
              I%(I) = EC(N,0):
      NEXT I

366   & O(EC,1 TO NE,0,A):
      HOME :
      RETURN
```

(368) Sort to identify nodes inside the plot boundaries.

```
368   PRINT "[M]SORTING NODES...":
      & W((NC,1 TO NN,1),SL,SR,N%,BR%,ER%):
      & W((NC,BR% TO ER%,2),ST,SB,N%,BR%,ER%):
      & O(NC,BR% TO ER%,0,A):
      FOR I = 1 TO N%:
          A = BR% +  I - 1:
          XL(I) = NC(A,1):
          YL(I) = NC(A,2):
          I%(I) = NC(A,0):
      NEXT I:
      NO = N%:
      HOME :
      RETURN

65535 REM [M][M]1FEB85[M]JRC/DCD

END-OF-LISTING
```

DISKETTE PREPARATION.ELASTICITY

Initialization.

```
1 CLEAR :
  TEXT :
  & D(T):
  WR$ = "...WHEN READY, PRESS <RETURN>":
  D$ = CHR$ (4):
  FI$ = "FILEINFO.TXT":
  POKE 2164,7:
  REM DISKETTEPREP(E)
```

Title.

```
2 HOME :
  GOSUB 40:
  HTAB 15:
  FLASH :
  PRINT "ELASTICITY":
  NORMAL :
  HTAB 4:
  PRINT "DATA DISKETTE PREPARATION (7.0)":
  GOSUB 40:
  PRINT "[M][M]
     ABSTRACT: STORE THE PROBLEM DESCRIPTION[M]
     AND DEFAULT DATA FILE INFORMATION."

4 DR = PEEK (2048):
  DR$(1) = ",D1":
  DR$(2) = ",D2":
  PRINT "[M]INSERT ";:
  FLASH :
  PRINT "DATA";:
  NORMAL :
  PRINT " DISKETTE TO BE PREPARED INTODRIVE "DR:
  & I("[M]<RETURN>";"";A$,"1"):
  ONERR GOTO 42

6 PRINT "IF FORMATTED,":
  & I("CATALOG? (Y/N)";"N";A$,"2",0,1):
  IF A$ = "Y" THEN
       PRINT D$"CATALOG";DR$(DR):
       & I("CONTINUE? (Y/N)";"Y";A$,"3",0,1):
       ON A$ < >"Y" GOTO 34
```

Format a data diskette having neither DOS nor greeting program.

```
8 PRINT D$"PREP"DR$(DR):
  POKE - 16368,0:
  HOME :
  & C(NA$,DE$,D1$,D2$):
  DIM NA$(24),DE$(24),D1$(24),D2$(24):
  PRINT "[M]SPECIFY:":
  & I(" KEYWORD:";"PROB 1";KW$,"4",BC%,8):
  ON BC% < >0 GOTO 2:
  HTAB 3:
  & I("[M] DESCRIPTION:","",DS$,"5",BC%,80):
  ON BC% < >0 GOTO 2

10 DATA "COORDINATES","COORDINATES OF INPUT DATA POINTS",
     "POINTS","DIRECTIONS","REGIONS",
     "NODE AND CONNECTIVITY DATA","REGION(S)",
```

```
      "NUMBERS EACH","ELEMENT NODES","ELEMENT NODE NUMBERS",
      "ELEMENTS","NODES EACH"

12  DATA "NODE COORDS","COORDINATES OF NODES","NODES",
      "DIRECTIONS","RENUM EL ND",
      "RENUMBERED ELEMENT NODE NUMBERS","ELEMENTS",
      "NODES EACH","RENUM ND COORDS",
      "COORDINATES OF RENUMBERED NODES","NODES",
      "DIRECTIONS"

14  DATA "RENUM LINES","RENUMBERED NODES FOR UNIQUE LINES",
      "LINES","NODES/EL"

16  DATA "RENUM BOUNDS","RENUMBERED BOUNDARY NODES & ELEMENTS",
      "NODES","NODES/EL","PROPERTIES",
      "ELASTIC & THERMAL PROPERTIES","ELEMENTS",
      "PROPERTIES","INPUT BC",
      "INPUT BOUNDARY CONDITION SPECIFICATIONS"

18  DATA "INPUTS","PARAMETERS","BOUND COND",
      "NODAL D.O.F. BOUNDARY CONDITIONS","D.O.F.",
      "CODE/VALUE","INIT FORCES",
      "INITIAL (THERMAL) FORCE VECTOR",
      "D.O.F.","VALUE","INIT STIFF",
      "INITIAL GLOBAL STIFFNESS MATRIX"

20  DATA "D.O.F.","BANDWIDTH","COMB FORCES",
      "COMBINED INITIAL AND BOUNDARY FORCES","D.O.F.",
      "VALUE","MOD FORCES",
      "FORCES MODIFIED BY DISPLACEMENT B.C.","D.O.F.",
      "VALUE","MOD STIFF",
      "STIFFNESS MODIFIED BY DISPLACEMENT B.C."

22  DATA "D.O.F.","BANDWIDTH","NODE DISP",
      "NODAL DISPLACEMENT MAGNITUDES","D.O.F.",
      "VALUE","EL STRAINS",
      "ELEMENT STRAINS IN COORD DIRECTIONS","ELEMENTS",
      "COMPONENTS","EL STRESSES",
      "ELEMENT STRESSES IN COORD DIRECTIONS"

24  DATA "ELEMENTS","COMPONENTS","EL P STRESS",
      "ELEMENT PRINCIPAL STRESSES & DIRECTIONS",
      "ELEMENTS","MAGN/DIR","NODE STRESS",
      "NODAL STRESSES IN COORD DIRECTIONS","NODES",
      "COMPONENTS"

26  DATA "ND PR STRESS","NODAL PRINCIPAL STRESSES & DIRECTIONS",
      "NODES","MAGN & DIR","NEW COORDS",
      "NEW COORDINATES OF NODES","NODES","DIRECTIONS",
      "ND REACTIONS","RESULTING REACTIONS AT NODES",
      "D.O.F.","VALUE"
```

Prepare the directory of data files.

```
28  PRINT "[M][M]CREATING "FI$:
    RESTORE :
    FOR I = 1 TO 24:
        READ NA$(I),DE$(I),D1$(I),D2$(I):
    NEXT I:
    N = 1:
    S = 0:
    PRINT D$;"OPEN"FI$",L100";DR$(DR):
    PRINT D$;"DELETE"FI$:
    PRINT D$;"OPEN"FI$",L100":
```

```
    PRINT D$;"WRITE"FI$",R0":
    PRINT N:
    PRINT KW$:
    PRINT DS$

30  FOR RE = 1 TO 24:
       PRINT D$"WRITE"FI$",R";RE:
       PRINT S:
       PRINT KW$ + "/" + NA$(RE):
       PRINT DE$(RE):
       PRINT N:
       PRINT D1$(RE):
       PRINT N:
       PRINT D2$(RE):
    NEXT RE:
    PRINT D$"CLOSE"FI$:
    PRINT "COMPLETED."
```

Review the directory file.

```
32  & I("REVIEW CONTENTS? (Y/N)";"N";A$,"6",0,1):
    IF A$ = "Y" THEN
        PRINT "[M]USE CTRL S TO STOP/START LISTING":
       & B(10,10):
       & B(0,0,10):
       PRINT D$"TDUMP"FI$;DR$(DR):
       & I(WR$;"";A$,"7")
```

(34-38)Exit menu.

```
34  HOME :
    PRINT
      "*** DISK PREP: EXIT (7.1) ***[M][M]
      1. RETURN TO MAIN MENU[M]
      2. PREPARE ANOTHER DISKETTE[M][M]
      0. STOP[M][M]":
    & I("SELECT 0-2 ";1;A,"8",0,1,A > = 0 AND A < = 2):
    ON A GOTO 36,2:
    END

36  POKE 2165,1:
    ON DR = 2 GOTO 38:
    PRINT "[M]INSERT ";:
    FLASH :
    PRINT "PROGRAM";:
    NORMAL :
    PRINT " DISKETTE INTO DRIVE 1":
    & I(WR$;"";A$,"9")

38  PRINT D$"RUN HELLO,D1"

40  FOR I = 1 TO 40:
       PRINT "*";:
    NEXT I:
    RETURN
```

(42-50)Error messages.

```
42  POKE 216,0:
    ER = PEEK (222):
    ON ER = 255 GOTO 50:
    ON ER = 4 GOTO 46:
    ON ER = 8 GOTO 44:
    PRINT "[G][G]ERROR #"ER:
    END
```

```
    44  PRINT "DOOR OPEN, I/O ERROR OR[M]NOT YET INITIALIZED)":
        GOSUB 48:
        RESUME

    46  PRINT "[G]WRITE PROTECTED":
        GOSUB 48:
        RESUME

    48  & B(20,20):
        & I(WR$;"";A$,"10"):
        RETURN

    50  END

65535 REM [M][M]1FEB85[M]JRC/DCD

END-OF-LISTING
```

TEXT.ELASTICITY

```
2  TEXT :
   & D(T):
   HOME :
   INVERSE :
   FOR I = 1 TO 4:
       PRINT  TAB( 2)" " TAB( 38)" ":
   NEXT I:
   NORMAL :
   VTAB 2:
   HTAB 3:
   PRINT " APPLIED FINITE ELEMENT ANALYSIS: ":
   HTAB 3:
   PRINT "     AN APPLE " CHR$ (221) CHR$ (219)
      " IMPLEMENTATION      ":
   VTAB 6:
   HTAB 12:
   PRINT "COPYRIGHT, 1985":
   PRINT

4  HTAB 9:
   INVERSE :
   PRINT "ELASTICITY TEXT";:
   NORMAL :
   PRINT " (8.00)":
   PRINT "ABSTRACT:":
      PRINT "GRID, PROPERTY,
      AND B.C. DATA ARE READ  FROM BINARY DATA FILES
      AND WRITTEN TO    TEXT FILES."

6  POKE 2166,8:
   FI$ = "FILEINFO.TXT":
   GOSUB 508:
   & C(DR$):
   DIM DR$(2):
   DR$(1) = ",D1":
   DR$(2) = ",D2":
   & B(200,20):
   PRINT "INSERT ";:
   FLASH :
   PRINT "DATA";:
   NORMAL :
   PRINT " DISKETTE INTO DRIVE "DR:
   GOSUB 508:
   ONERR  GOTO 510

8  PRINT D$"OPEN"FI$",L100"DR$(DR):
   PRINT D$"READ"FI$",R0":
   INPUT TE,KW$,PD$:
   PRINT D$"CLOSE"FI$:
   PRINT "PROBLEM KEYWORD: "KW$:
   PRINT :
   PRINT "PROBLEM DESCRIPTION :":
   PRINT PD$:
   GOSUB 508:
   & C(SI%):
   DIM SI%(22)

10 RE = 5:
```

```
          GOSUB 500:
          NE = N1:
          DE = N2:
          & C(ND%):
          DIM ND%(NE,DE):
          & R(ND%,N$):
          GOSUB 506:
          RE = 6:
          GOSUB 500:
          NN = N1:
          & C(XY):
          DIM XY(NN,2):
          & R(XY,N$):
          GOSUB 506:
          REM  " RETRIEVE NODES OF ELEMENTS (ND%)
             AND COORDINATES OF NODES (XY)

      12  RE = 7:
          GOSUB 500:
          NL = N1:
          & C(LN%):
          DIM LN%(NL,3):
          & R(LN%,N$):
          GOSUB 506:
          RE = 8:
          GOSUB 500:
          NB = N1:
          & C(BN%):
          DIM BN%(NB,2):
          & R(BN%,N$):
          GOSUB 506:
          REM " RETRIEVE NODES OF LINES (LN%) AND BOUNDARY NODES (BN%)

      14  RE = 9:
          GOSUB 500:
          & C(MP):
          DIM MP(NE,N2):
          & R(MP,N$):
          GOSUB 506:
          REM  " RETRIEVE MATERIAL PROPERTIES OF ELEMENTS (MP)

      16  REM " COL. 1 IS ELASTIC MODULUS;
             COL. 2 IS POISSON'S RATIO;
             COL. 3 IS THICKNESS;
             COL. 4 IS TEMPERATURE;
             COL. 5 IS THERMAL EXPANSION COEFFICIENT.
             MP(0,0) IS REFERENCE TEMPERATURE.
             ROW 0 IS PROPERTY UNIFORMITY (1=UNIFORM; 0=NONUNIFORM).

      18  RE = 11:
          GOSUB 500:
          NP = N1:
          & C(BC):
          DIM BC(NP,2):
          & R(BC,N$):
          GOSUB 506:
          REM  " RETRIEVE BOUNDARY CONDITIONS FOR EACH
             DEGREE OF FREEDOM.
             COL. 1 IS B.C. TYPE (1=FORCE, 2=DISPLACEMENT)

      20  PRINT :
          PRINT "------ DATA FILES HAVE BEEN READ ----- ":
          PRINT :
          PRINT "  PREPARE TO TRANSFER":
```

```
      END :
      REM "[M][M]ADD CODE TO OUTPUT TO TEXT FILE HERE[M][M]

  500 PRINT D$"OPEN"FI$",L100"DR$(DR):
      PRINT D$"READ"FI$",R"RE:
      INPUT SI%(RE),NA$,DE$,N1,D1$,N2,D2$:
      PRINT D$"CLOSE"FI$:
      ON N$ = NA$ + DR$(DR):
      ON SI%(RE) = 0 GOTO 510

  502 PRINT "*** ";:
      INVERSE :
      PRINT "LOADING";:
      NORMAL :
      PRINT " DATA FROM DISKETTE ***":
      PRINT :
      PRINT "FILE NAME FOR":
      & I(DE$,NA$,NA$,"1",BC%,30):
      ON BC% >0 GOTO 502:
      ON LEFT$ (A$, LEN (KW$)) = KW$ GOTO 504:
      NA$ = KW$ + "/" + NA$

  504 N$ = NA$ + DR$(DR):
      RETURN

  506 PRINT :
      PRINT DE$:
      PRINT "FOR "N1" "D1$" AND "N2" "D2$:
      PRINT "WERE LOADED":
      RETURN

  508 PRINT :
      & B(20,20):
      VTAB 24:
      & I("... WHEN READY, PRESS <RETURN>";"2";A$):
      HOME :
      RETURN

  510 PRINT :
      FLASH :
      PRINT CHR$ (7) CHR$ (7)"DATA NOT DEFINED":
      NORMAL :
      PRINT "EITHER FILE":
      INVERSE :
      PRINT NA$:
      NORMAL :
      PRINT "DOESN'T EXIST OR THE DATA IS NOT VALID":
      END

65535 REM [M][M]1FEB85[M]JRC/DCD

END-OF-LISTING
```

3.4.1 Variable Definitions and Cross References

```
*******************************
*                             *
*      HELLO.ELASTICITY       *
*                             *
*     TABLE OF VARIABLES      *
*                             *
*******************************
```

A - GENERAL VARIABLE; FREQUENTLY ASCII VALUE OF INPUT CHARACTER
12 12 12 12 12 12 12 12 14
14 14 14 16

A$ - GENERAL INPUT VARIABLE
10 12 16 16 20 38

B(*,*) - & BEEP COMMAND
12 12 30

B(*,*,*) - & BEEP (WITH REPETITION)
30

BC% - BRANCH CODE (1 ='<' BACKWARD; 2 ='>' FORWARD BRANCH; 3 = ':' A PREVIOUS VALUE
14 14 20 20

D$ - CHR$(4) CONTROL D
4 6 8 8 36 36

D(*) - & DISPLAY (&D(T)=TEXT,&D(G)=MIXED,&D(V)=FULL GRAPHICS)
6

DR - DISKETTE DRIVE NUMBER
36

DR$(*) - DISKETTE DRIVE STRING
36

I - GENERAL INDEX
18 18 26 26 26 26 26 26 26
28

I(*,*) - & INPUT CMD
10 12 38

I(*,*,*) - & INPUT CMD
20

I(*,*,*,*) - & INPUT CMD
16

I(*,*,*,*,*) - & INPUT CMD
12 28

I(*,*,*,*,*,*) - & INPUT CMD
14

LI(*) - LINE LOCATION IN HELLO MENU
32

N - MENU OPTION
26 28 28 28 28 28 32 32 32
32 32 32 36

PD$(*) - PROBLEM DESCRIPTOR
26 26 26 26

PN$(*) - PROGRAM NAMES
26 32 36

SR$ - 'SELECT A NUMBER' PROMPT
4 14 28

T - &D(T) SETS TEXT DISPLAY
6

WR$ - 'WHEN READY' PROMPT
4 10 12 20

1 FEB 85 Version JRC/DCD

END OF VAR. LIST

```
*******************************
*                             *
*    GEOMETRY.ELASTICITY      *
*                             *
*     TABLE OF VARIABLES      *
*                             *
*******************************
```

See the identical GEOMETRY.HEAT table in Chapter 2, page 206.

```
*******************************
*                             *
*      GRID.ELASTICITY        *
*                             *
*     TABLE OF VARIABLES      *
*                             *
*******************************
```

See the identical GRID.HEAT table in Chapter 2, page 213.

```
******************************
*                            *
*   PREPROCESS.ELASTICITY    *
*                            *
*     TABLE OF VARIABLES     *
*                            *
******************************
```

*CX(*) - CALCULATES X COORDINATE OF ELEMENT CENTROID
60 244 308 312

*CY(*) - CALCULATES Y COORDINATE OF ELEMENT CENTROID
60 244 308 312

*M(*) - STEPPING FUNCTION AROUND BOUNDARIES
10 24 26 32 34 46 46 52 56
240 330 330

*N(*) - STEPPING FUNCTION FOR N
20 22 28 240

*P2(*) - STEPPING FUNCTION (CCW) AROUND AN ELEMENT
60 240

*PX(*) - CONVERTS X ABSOLUTE COORDINATE TO SCREEN COORDINATE
10 10 50 52 56 56 60 60 174
244 248 306 306

*PY(*) - CONVERTS Y ABSOLUTE COORDINATE TO SCREEN COORDINATE
10 10 50 52 56 56 60 60 174
244 248 306 306

*UX(*) - CONVERTS X SCREEN COORDINATE TO ABSOLUTE COORDINATE
248

*VY(*) - CONVERTS Y SCREEN COORDINATE TO ABSOLUTE COORDINATE
248

A - GENERAL VARIABLE; FREQUENTLY ASCII VALUE OF INPUT CHARACTER
10 24 28 30 30 34 46 46 52
56 98 98 98 98 98 98 98 98
100 100 100 100 100 100 100
102 102 102 102 106 106 106
106 106 106 106 106 108 108
122 122 122 122 122 122 122
122 124 124 124 124 124 124
124 124 124 126 126 126 126
176 176 176 176 180 184 184
240 240 306 330 330

A$ - GENERAL INPUT VARIABLE
4 10 10 10 10 20 22 24 44
44 98 98 98 100 104 104 122
122 122 124 124 134 134 134
134 134 134 134 138 138 138
138 138 138 138 142 142 142
142 142 142 142 144 146 146
146 146 146 146 152 152 152
152 152 152 154 154 154 154
156 156 224 224 226 236 266
274 274 278 278 280 280 310
312

A0 - GENERAL VARIABLE
12 12 48 48 48 54 54 54 102
102 102 102 126 126 126 126
152 152

AC - ACTION OPTIONS(1=ENTER;2=DELETE;3=QUIT)
20 22 24 26 32 134 136 140
144 146 148 152

AC$(*) - STRING FOR ACTION OPTION (E=ENTER;D=DELETE;Q=QUIT)
136 140 144 146 152 246 254
254

B - BOUNDARY NUMBER
10 24 26 32 34 46 46 52 56
56 56 56 56 56 56 62 62 62
62 64 132 132 132 154 154 154
154 154 240 328 330 330

B(*,*) - & BEEP COMMAND
4 6 98 122 236 266

B(*,*,*) - & BEEP (WITH REPETITION)
6

B1 - GENERAL VARIABLE
46 46 56 56 330 330

B2 - GENERAL VARIABLE
46 46 56 56 330 330

BC - EQUIVALENT BOUNDARY CONDITION MATRIX
164 168 170 318 320 324 330

BC% - BRANCH CODE (1 ='<' BACKWARD; 2 ='>' FORWARD BRANCH; 3 = ':' A PREVIOUS VALUE
4 74 74 76 78 78 78 80 80
82 84 86 88 90 92 94 94 94
94 98 98 98 100 100 100 100
102 104 104 106 108 108 112
116 122 122 122 124 124 124
126 132 134 134 138 138 142
142 146 146 148 148 152 152
154 180 182 188 190 192 198
198 206 206 258 258 262 262
266 266 274 274 274 280 280
290 290 292 298 298 300 300
302 302 310 312

BC(*,*) - EQUIVALENT BOUNDARY CONDITION MATRIX
36 36 36 36 36 36 38 38 38
38 38 38 38 38 38 38 38 38
40 40 40 42 42 164 318 324
324 330 330

BD - CODE FOR BOUNDARY CONDITION DIRECTION (1=X-DIRECTION;2=Y-DIR;3=NORMAL TO BOUNDARY (+OUT);4=NORMAL TO FACE (SCREEN))
12 16 144 144 146 146 148

BD$(*) - STRING FOR BD
142 142 142 142 144 146 146
146 148 246 254 254 254 254
316

BI - NODE # FOR NODE RENUMBERING
158 160 162 284 286 288 316

BI% - MATRIX OF INPUT BOUNDARY CONDITION CODES
288 290 290 326

BI%(*,*) - MATRIX OF INPUT BOUNDARY CONDITION CODES (COL 1: INDEX IN BN%, COL 2 : BC TYPE; COL 3 DIRECTION)
12 12 12 14 14 14 16 16 16
34 34 40 40 42 42 46 46 46
64 158 164 164 164 288 288
290

BI(*,*) - COMBINED BI% AND BV
158 158 158 284 288 288 316
316 316 316

BN% - BOUNDARY NODES
188 190

BN%(*,*) - BOUNDARY NODES ARRAY WITH NODES IN ORDER AROUND BOUNDARIES
10 12 16 34 34 34 40 42 46
46 46 52 56 56 172 174 188
316 324 324 328 330 330

BR% - BEGINNING ROW NUMBER IN &W()
308 308

BT - TYPE OF B.C. (0=FREE; 1=FORCE; 2=DISPLACEMENT; 3=SURFACE STRESS)
12 16 46 46 46 46 48 48 48
48 48 50 50 138 140 140 144
146 148 148 150 330 330 330

BT$(*) - STRING FOR BT
138 138 138 140 144 146 148
148 246 252 252 252 252 316
324

BV - BOUNDARY VALUE
288 290 290 326

BV(*) - BOUNDARY VALUE
14 14 14 16 36 40 42 46 158
288 288 290

C(*) - CLEAR ARRAY (& C(*))
158 162 164 172 178 180 184
186 188 240 264 272 284 288
290 298 304 316 318 324 330

C(*,*) - & CLEAR ARRAY CMD
170 234 274 278 288 290 290
302 326

C(*,*,*) - & CLEAR ARRAY CMD
180

CO%(*) - CODES FOR COLOR OF FILL
60 246 308

CR - COORDINATES OF INPUT REGION CENTROID
172

CR(*,*) - COORDINATES OF REGION CENTROIDS
120 120 172 172 172 172 172
172 172 174 174 174 174 174
174

CV(*,*) - CODED VALUES FOR MATERIAL PROPERTIES
90 90 90 90 92 92 100 118
124 246 248 248 248

D$ - CHR$(4) CONTROL D
1 44 196 196 196 200 200 200
204 214 214 214 216 216 216
224 268 276 276 276 276 276
290 290 290

D(*) - & DISPLAY (&D(T)=TEXT,&D(G)=MIXED,&D(V)=FULL GRAPHICS)
56 102 126 130 156 228 258
274 280 292 304 316 324

D1 - LENGTH OF DIAGONAL ACROSS QUADRILATERAL
36 36 36 36 38 38 38 38 38
38 38 38

D1$ - ROW DESCRIPTOR FOR SAVED DATA
196 200 200 212

D2 - LENGTH OF DIAGONAL ACROSS QUADRILATERAL
36 36 36 36 38 38 38 38 38
38 38 38 38 38

D2$ - COLUMN DESCRIPTOR FOR SAVED DATA
196 200 200 212

DB - DIMENSION OF BOUNDARY VALUE ARRAY
188 288 288 290 290

DE - # NODES / ELEMENT
60 172 178 178 240

DE$ - DESCRIPTION OF DATA IN FILEINFO.TXT
196 198 200 200 206 212

DF - DEGREE OF FREEDOM
40 40 40 40 42 42 42 330 330 330

DF$(*) - COORDINATE (DIRECTION) DESCRIPTOR
246 254 254 254 254 254 254

DI - DIRECTION CODE
34 36 36 36 38 38 38 38 40 40 42 42 46 46 48 48 48 48 52 52 52 52 52 52 330 330 330 330

DI$(*) - GENERAL COORDINATE (DIRECTION) DESCRIPTOR
58 58 246 254 254 254 254 324

DR - DISKETTE DRIVE NUMBER
44 196 196 200 204 208 214 216 224 234 234 266 276 290

DR$ - DISKETTE DRIVE STRING
234

DR$(*) - DISKETTE DRIVE STRING
44 196 196 200 204 208 214 216 224 234 234 234 276 290

E$ 'ENDING'
1 24 30

E0 - ELEMENT NUMBER
84 106 106 106

E1 - ELEMENT NUMBER
72 96 106

E2 - ELEMENT NUMBER
72 96 102

E3 - ERROR FLAG FOR DISK I/O
204 204 226

ED - EDIT FLAG (1=EDIT MODE; 0=NOT)
70 114 114 128 132 156 156 274 274 280 290 290

EF - ERROR FLAG (1=ERROR OCCURRED; 0=NONE)
160 162 168 168 180 180 182 182 188 188 190 190 218 272 272 276 276 286 286 320 320

EL - ELEMENT # (GENERAL)
34 34 60 60 60 60 60 84 84 84 96 96 96 98 100 100 100 100 102 102 102 106 106 106 106 120 120 120 172 172 172 172 244 244 244 244 244 244 244 244 294 294 294 294 308 308 308

EO - EXIT OPTION (1=PROCEED TO NEXT PROGRAM; 2=TO MAIN MENU; 3=REMAIN IN SAME; 0=STOP)
96 262 262 262 262 262 266 266 268 268 268 268 268 268 268

ER - ERROR CODE
218 218 218 218 218 218 218 218 218 218 218 218 218 218 224 224 224 224 224 224 224

ER$ - ERROR CODE STRING DESCRIPTORS
234

ER$(*) - ERROR CODE STRING DESCRIPTORS
224 234 234 234 234 234 234 234

ER% - ENDING ROW NUMBER RETURNED FROM A VARIABLE SEARCH
308 308 308 308

F - &F FILL; COUNTER OF # BOUNDARY CONDITIONS ON THE FACE OF ELEMENTS (AS SEEN ON SCREEN)
36 36 36 38 38 38 38

F(*,*,*,*) - AMPERSAND FILL COMMAND
60 308 312

FI$ - 'FILEINFO.TXT' STRING
1 196 196 196 200 200 200 204 214 214 214 216 216 216

G - PARAMETER IN &D(G) SETS MIXED TEXT AND GRAPHICS
56 304

I - GENERAL INDEX
14 14 14 14 14 14 54 54 54 54 54 54 60 60 60 60 70 70 70 70 90 90 90 90 90 90 92 92 94 94 94 94 112 112 112 112 118 118 118 118 118 118 118 118 118 126 126 126 126 158 158 158 158 158 158 174 174 174 174 184 184 184 184 184 184 228 228 240 240 240 240 248 248 248 248 248 276 276 276 288 288 288 288 288 288 290 290 290 300 300 300 300 306 306 306 306 308 308

308 310 310 310 310 310 310
310 310 310 310 310 310 310
316 316 316 316 316 316 324
324 324 324 332 332 332 332

I(*,*) - &INPUT COMMAND
226 236

I(*,*,*) - & INPUT CMD
4 74 80 92 98 122 148 266
310 312

I(*,*,*,*) - & INPUT CMD
44 104 154 156 224 274

I(*,*,*,*,*) - & INPUT CMD
82 86 90 106 134 138 142 146
152 278 280

I(*,*,*,*,*,*) - & INPUT CMD
76 104 112 198 206 258 262
300

I1 - STARTING SUBSCRIPT FOR A BOUNDARY
56 56 56 56 62 62 132 154
240 240 240 240 240 240 240

I2 - ENDING SUBSCRIPT FOR A BOUNDARY
56 56 56 56 62 62 132 154
240 240 240 240 240 240

IO - INPUT OPTION (1=REGION; 2=ELEMENT)
60 76 76 76 80 86 86 86 86
86 86 96 104 104 104 104 106
106 106 106 106 106 106

IO$(*) - INPUT OPTION DESCRIPTOR
76 76 86 104 104 106 250 250
250

IS(*) - ARRAY FOR START & END SUBSCRIPTS FOR BOUNDARIES
56 56 62 62 132 132 154 154
174

J - GENERAL SUBSCRIPT
14 14 14 14 14 14 14 158 158
158 158 288 288 288 288 306
308 308 324 324 324 324

J1 - GENERAL SUBSCRIPT
66 66 66 66 68 68 68 68

K - GENERAL SUBSCRIPT
324 324 324

KW$ - KEYWORD
210 210 210 214 214 240

L - LINE LENGTH
8 8 8 36

L$ - GENERAL LABEL STRING
60 66 66 66 68 68 102 120
126 126

L(*) - & LISTING CONTROL
1 294 298 316 316 324 324

LN% - LINE NODE NUMBER ARRAY
186 188

LN%(*,*) - LINE NODE NUMBER ARRAY
186 306 306

MO - MENU OPTIONS
258 258 258 258 258 258 278
292 292 292 292

MP - MATERIAL PROPERTY ARRAY
108 108 272 272 274 276 278
298 302 306 308

MP$ - MATERIAL PROPERTY DEFAULT
96 96 98

MP(*,*) - MATERIAL PROPERTY ARRAY
70 74 74 80 80 82 82 82 82
84 84 84 96 100 108 108 108
108 126 272 274 276 294 294
296 296 304 308 308 312 332
332 332

MU$(*) - MENU (2.05) OPTION STRING (GRID)
240 242 242 242 242 242 242
244 244 244 244 256 256 256
256 256 256 256 256 274 280
292

MX - MAX. OF MC AND MR
106 106

N - SUBSCRIPT; COMMONLY IN STEPPING FUNCTION
10 10 10 12 12 16 16 20 20
22 22 22 24 24 24 24 24 26
26 26 28 28 28 30 30 32 32
32 34 34 34 34 40 40 42 42
46 46 46 46 52 56 56 56 56
62 62 64 172 172 172 196 240
240 240 240 240 240 240 240
240 240 328 328 330 330 330

N$ - GENERAL STRING
160 168 180 180 182 188 190
196 208 272 276 286 320

N0 - GENERAL INDEX
34 34 56 56

N1 - GENERAL INDEX; NUMBER OF ROWS IN SAVED ARRAY
22 24 24 24 28 30 30 40 40
42 42 60 60 60 158 166 172
172 172 178 180 180 180 186
188 196 200 200 212 272 274
284 306 306 306

R0 - REGION #
84 106 106 106 120

R1 - REGION # OR RECORD NUMBER
72 106 120 170 216 278

R2 - REGION # OR RECORD #
72 86 86 86 120 126

RE - REGION # OR RECORD #
60 60 106 106 106 106 120 120
120 120 120 120 120 122 124
124 124 124 126 126 126 126
158 166 172 172 172 172 172
172 172 174 174 174 174 174
174 174 174 178 178 180 180
186 186 188 188 196 196 200
200 200 216 216 216 216 270
270 274 280 284 284 318 318

RP - REGIONAL PROPERTIES
274 278

RP$ - REGION PROPERTY DEFAULT
120 120 122

RP(*,*) - REGIONAL MATERIAL PROPERTIES
120 124 126 274

S$ - 'STARTING'
1 22 28

S(*,*) - & SAVE ARRAY CMD
160 168 276

SB - SCREEN BOTTOM Y COORDINATE
176 248 248 248

SC - SCALE
176 176 176 176 176 248 248
248 248

SI%(*) -FILE STATUS INDICATOR
(0=UNDEFINED)
178 180 186 188 196 200 200
216 240 270 280 284 318

SL - SCREEN LEFT X COORDINATE
176 248 248 248

SR - SCREEN RIGHT X COORDINATE
176 248

SR$ - 'SELECT A NUMBER' PROMPT
1 76 104 112 258 262 300

ST - NODAL STRESSES
176 248

SY - SYMBOL NUMBER
46 48 50 54 54 330

SZ - SIZE INDICATOR FOR PLOTTING B.C. SYMBOLS
46 64 64 128 328 330 330

T - &D(T) SETS TEXT DISPLAY
102 126 130 156 228 258 274
280 292 316 324

T1 - TYPE OF MATERIAL PROPERTY
70 72 78 112 112 112 112 114
114 300 302

T2 - TYPE OF MATERIAL PROPERTY
70 72 114 300 302

TE - TYPE OF ELEMENT (1=TWO-DIMENSIONAL; 2=AXISYMMETRIC)
34 34 36 36 36 36 72 112 214
254 254 294 300 302

TF$ - TEMPORARY FILENAME FOR THICKNESS DATA
240 276 276 276 276 276 290
290 290

TH - THICKNESS OF ELEMENT
34 36 170 290

TH(*) - THICKNESS OF ELEMENT
34 290 290

TY - TYPE OF MATERIAL PROPERTY (1=ELASTIC MODULUS; 2=POISSON'S RATIO; 3=THICKNESS; 4=TEMPERATURE; 5=THERMAL EXPANSION COEFFICIENT) (E)
72 72 72 72 74 76 78 78 78
78 78 80 80 80 80 80 82 82
82 82 82 84 84 84 86 86 88
88 90 90 90 90 90 92 92 92
96 98 100 100 104 108 108 108
108 108 108 108 108 108 110
112 112 112 112 118 120 122
124 124 126 126 294 294 294
294 294 294 294 294 294 296
296 296 296 298 298 298 300
300 300 300 300 300 300 300
300 300 300 302 302 302 302
302 302 302 304 304 304 306
308 308 312 312

TY$(*) - DESCRIPTOR OF MATERIAL PROPERTY TYPES
72 72 76 80 80 86 86 88 98
104 108 108 112 122 246 252
252 252 252 252 294 294 294
296 296 300 304 304 312

U$ - STRING FOR USE OF KEYS IN MOVING LABELS
1 20 22 24 28 30

UH - HIGH VALUE OF X COORDINATE IN SPECIFIC PLOT
176

UL - LOW VALUE OF X COORDINATE IN SPECIFIC PLOT
176 248 248

UN - UNIFORMITY OF MATERIAL PROPERTY FLAG (0=NOT, 1=UNIFORM)
80 80 84 84

UN$ -UNIFORMITY DESCRIPTOR
240

UN$(*) - UNIFORMITY DESCRIPTOR
108 240 240

VB - BOUNDARY VALUE
16 46 46 48 48 54 54 148 148
330 330

VH - HIGH VALUE OF Y COORDINATE IN SPECIFIC PLOT
176

VL - LOW VALUE OF Y COORDINATE IN SPECIFIC PLOT
176 248 248

W(*,*,*,*,*,*) - AMPERSAND 'WHICH' COMMAND; SELECT VALUES WITHIN SPECIFIED RANGE
308

W1 - WEIGHTING FACTOR
36 36 38 38

W2 - WEIGHTING FACTOR
36 36 38 38

WR$ - 'WHEN READY' PROMPT
1 4 226 266

X - FUNCTION DUMMY ARGUMENT
248 248 248 248

X1 - GENERAL X COORDINATE; START OF LINE
8 8 8 36 36 52 52 54 60 60

X2 - GENERAL X COORDINATE; END OF LINE
8 8 8 36 36 52 60 60

XH - MAX X COORDINATE
184

XL - MIN X COORDINATE
184

XN - X COORDINATE OF NODE
180

XN(*) - X COORDINATE OF NODE
8 8 10 10 50 56 56 60 60
172 180 184 244 244 244 306
306

XP - GENERAL SCALED X COORDINATE
60 60 66 68 68 120

XY - COORDINATES OF POINTS
180 182 184 184 184

XY(*,*) - COORDINATES OF POINTS
180 184 184 184 184 184 184

Y - GENERAL Y COORDINATE
248 248 248 248

Y1 - GENERAL Y COORDINATE; START OF LINE
8 8 8 52 52 54 60 60

Y2 - GENERAL Y COORDINATE; END OF LINE
8 8 8 52 60 60

YH - MAX Y COORDINATE
184

YL - MINIMUM Y COORDINATE
184

YN - Y COORDINATE OF NODE
180

YN(*) - Y COORDINATE OF NODE
8 8 10 10 50 56 56 60 60
172 180 184 244 244 244 306
306

YP - GENERAL SCALED Y COORDINATE
60 60 66 68 68 120

Z - USER INTERACTION INDICATOR (0=MAX; 1=INTERMEDIATE; 2=FREE RUN; 3=DEMO)
6 112 156 198 198 206 212 238
256

1 FEB 85 Version JRC/DCD

END OF VAR. LIST

```
******************************
*                            *
*     SOLVE.ELASTICITY       *
*                            *
*    TABLE OF VARIABLES      *
*                            *
******************************
```

*P1(*) - STEPPING FUNCTION (CW) AROUND ELEMENT
38 228

DK - DIMENSION OF ELEMENT STIFFNESS MATRIX
42 44 44 44 44 44 44 44 52
52 54 56 56 220

DR - DISKETTE DRIVE NUMBER
172 176 182 188 194 198 206
210 216 216 240

DR$ - DISKETTE DRIVE STRING
216

DR$(*) - DISKETTE DRIVE STRING
172 176 182 188 194 198 206
210 216 216 216

DT - DIFFERENCE BETWEEN ELEMENT TEMPERATURE AND REFERENCE TEMPERATURE
36 36 40

E0 - INITIAL THERMAL STRAIN
36 44 50

E0(*,*) - INITIAL THERMAL STRAIN
36 44

E2 - ERROR FLAG
182 182 208

EF - ERROR FLAG (1=ERROR OCCURRED; 0=NONE)
64 66 68 70 78 78 90 92 100
102 106 108 120 122 124 126
134 134 136 136 146 148 152
152 164 164 166 166 200 224
224 226 226

EL - ELEMENT # (GENERAL)
8 28 28 28 28 30 30 30 30
30 30 30 30 34 36 36 36 40
46 46 48 48 48 48 50 52 54
54 60

EN% - NODE # OF ELEMENT
164

EN%(*,*) - ELEMENT NODE #S (ELEMENT#, LOCAL NODE #)
34 220 220

EO - EXIT OPTION (1=PROCEED TO NEXT PROGRAM; 2=TO MAIN MENU; 3=REMAIN IN SAME; 0=STOP)
230 238 238 238 240 240 240
242

ER - ERROR CODE
200 200 200 200 200 200 200
200 200 200 200 200 200 200
206 206 206 206 206 206 206

ER$ - ERROR CODE STRING DESCRIPTORS
216

ER$(*) - ERROR CODE STRING DESCRIPTORS
206 216 216 218 218 218 218
218

FE - ELEMENT FORCE MATRIX
44 50 50 50

FE(*,*) - ELEMENT FORCE MATRIX
44 54 56

FI$ - 'FILEINFO.TXT' STRING
1 172 172 172 176 176 176 182
194 194 194 198 198 198

FR - FREE RUN FLAG (1=FREE RUN; 0=USER CONTROL)

4 4 4 4 10 46 80 110 138
154 174 178 178 178 184 190
230 230 240 246 246 246 246
246 250

G1 - PIVOT VALUE FOR GAUSSIAN ELIMINATION
14 16 16

G2 - STIFFNESS VALUE
16 16 16

GF - GLOBAL FORCE MATRIX
44 64 78 90 100 120 134

GF(*) - GLOBAL FORCE MATRIX
16 16 16 20 20 22 22 24 44
56 56 82 82 142 142 156 156
220

GS - GLOBAL STIFFNESS MATRIX
44 68 106 124 136

GS(*,*) - GLOBAL STIFFNESS MATRIX
14 16 16 16 16 20 20 22 22
24 24 24 24 24 44 56 56 56
142 142 142 158 158 220

H - HORIZONTAL INDEX
12 12 12 12 12 12 46 140

H1 - HORIZONTAL INDEX
12 12 12 12 12

I - GENERAL INDEX
14 14 14 14 14 14 16 16 16
16 16 30 30 30 30 32 32 32
32 32 32 32 32 32 32 32 32
32 34 34 34 34 34 34 34 34
34 34 34 34 34 34 38 38 38
38 38 38 38 40 40 40 40 40
42 42 42 42 42 52 52 52 52
52 54 54 54 54 56 56 56 56
58 82 82 82 82 82 82 142 142

142 142 142 142 142 142 142
154 154 154 154 154 156 156
156 156 158 158 158 158 158
160 160 160 212 212 228 228
228 228 228 228

I(*,*,*) - & INPUT CMD
4

I(*,*,*,*) - & INPUT CMD
74 96 130 138 206 210

I(*,*,*,*,*) - & INPUT CMD
6

I(*,*,*,*,*,*) - & INPUT CMD
8 80 112 174 184 228 236 228
246

J - GENERAL SUBSCRIPT
16 16 16 16 16 16 20 20 20
20 22 22 24 30 30 30 32 32
32 32 32 32 32 38 38 38 38
38 38 38 38 38 38 40 42 42
42 42 56 56 56 58 142 142
142 142 158 158 158 158

K - GENERAL SUBSCRIPT
16 16 16 16 16 20 22 22 22
22 22 22 22 36 36 38 38 38
38 38 38 38 52 52 52 52 52
52 52 52 52 52 56 56 56 56
142 142 142 160 160

K2 - GENERAL INDEX
160 160

KE - ELEMENT STIFFNESS MATRIX
44 50 50 50

KE(*,*) - ELEMENT STIFFNESS MATRIX
44 52 52 56

KI - GENERAL INDEX
56 56 56 56 56 56 56 56 56

KJ - GENERAL INDEX
56 56 56 56

KW$ - KEYWORD
186 186 186 194 196 196

L(*) - & LISTING CONTROL
1 32 32 40 42 52 54 58 154
154 156 156 158 158 160 160

LA$ - GENERAL LABEL
82 82 84 114 114 116 116

M - SUBSCRIPT
20 20 20 20

M(*) - & MATRIX CMD
28 36 36 50 50 50 50 50 50

MJ - SUBSCRIPT
14 14 14 14 14 16 142 142

MK - SUBSCRIPT
14 14 16 16 16

ML - LABEL FLAG
82 84 114 114 116 116

MO - MENU OPTIONS
6 230 236 236 236 236 236

MP - MATERIAL PROPERTY ARRAY
224

MP(*,*) - MATERIAL PROPERTY ARRAY
28 28 28 28 30 30 30 30 30
30 30 30 36 36 36 36 50 220
224 224 224 224 224

N - SUBSCRIPT; COMMONLY IN STEPPING FUNCTION
14 16 16 16 16 142 142

N$ - GENERAL STRING
64 68 78 90 100 106 120 124
134 136 146 152 164 166 188
224 226

N1 - GENERAL INDEX; NUMBER OF ROWS IN SAVED ARRAY
62 66 78 88 102 108 118 122
134 136 144 150 152 172 172
176 178 190 220 220

N2 - GENERAL INDEX; NUMBER OF COLUMNS IN SAVED ARRAY
62 66 88 108 118 122 136 144
172 176 178 190 220

NA$ - FILENAME
168 172 174 174 176 178 184
184 186 186 186 188 190 192
206

NC - NODE COORDINATES
166

NC(*,*) - COORDINATES OF NODES (ABSOLUTE OR SCREEN COORDINATES)
34 34 220

ND - NODE #
44

ND(*) - NODE NUMBERS FOR AN ELEMENT
34 34 34 34 34 40 44

NE - NUMBER OF ELEMENTS
48 48 220 220 220

NM - NEXT MENU DEFAULT
230 230 236

NN - NUMBER OF NODES
220 220 220

NP - NUMBER OF DEGREES OF FREEDOM
14 14 14 14 14 14 14 20 20
20 44 44 62 66 78 82 88 102
108 118 122 134 136 142 142
142 142 142 142 142 142 142
144 152 154 156 158 220 220
220 220

NS - DEGREE OF FREEDOM FOR NODE AND DIRECTION
44

NS(*) - DEGREE OF FREEDOM FOR NODE AND DIRECTION
34 34 44 52 52 52 54 56 56

P% - PARAMETER FOR LISTING SPEED CONTROL
1 54 154 156 158 160

P2 - TWO*PI
50 226

PD$ - PROBLEM DESCRIPTOR
194 196

PN$ - PROGRAM NAMES

PN$(*) - PROGRAM NAMES
226 226 226 240 242

PO - PRINT OR PLOT OPTION
8 8 8 8 32 40 44 48 48 52
54 54 56 56 58 58 58 58 58
58 60 80 80 80 80 82 82 82
82 110 112 112 112 114 114
114 114 116 116 116 116 116
116 138 138 142 142 142

PS - PLANE STRESS/PLANE STRAIN INDICATOR (1=PLANE STRESS; 2=PLANE STRAIN)
28 30 36 144 228 228 228

R - RATIO OF MATERIAL PROPERTIES
28 28 28 30 30 30 32

R(*,*) - & RECALL ARRAY
78 100 106 134 136 152 164
166 224 226

R1 - RECORD NUMBER
70 92 126 148 198

R2 - RECORD NUMBER
70 92 126 148 198

RC - CENTROID OF ELEMENT (E)
34 34 34 34 34 38 38 50

RE - REGION # OR RECORD #
62 66 72 72 88 94 94 104 104
118 122 128 128 128 134 134
144 150 150 150 162 162 164
164 172 172 176 176 176 198
198 198 198 220 220 222 224

S(*,*) - & SAVE ARRAY CMD
64 68 90 120 124 146

SI%(*) - STATUS INDICATOR (0=INVALID DATA FILE; 1=ACTIVE)
72 94 104 128 128 134 150 150
162 164 172 176 176 198 220

SR$ - 'SELECT A NUMBER' PROMPT
1 8 80 112 228 236 238 246

SU - SUM IN MATRIX MULTIPLICATION
142 142 142 142

T - &D(T) SETS TEXT DISPLAY
1

TE - TYPE OF ELEMENT (1=TWO-DIMENSIONAL; 2=AXISYMMETRIC)
28 30 30 36 38 50 50 194 220
220 222 222 224 228

TRN(*) - TRANSPOSE MATRIX OPERATOR
50

U$ - UNDERLINE STRING
154 218 218

UN - UNIFORMITY OF MATERIAL PROPERTY FLAG (0=NOT, 1=UNIFORM)
46 50 224

UU - DISPLACEMENT MATRIX
142 146 150 152

UU(*) - DISPLACEMENTS OF ALL NODES
142 142 142 142 144 150 154
154 160 160

V - VERTICAL INDEX
12 12 12 12 12 12 46 140

VE - VOLUME OF ELEMENT
50 50 50

WR$ - 'WHEN READY' PROMPT
1 4

X - X COORDINATE MATRIX; DUMMY VARIABLE
44 222 222

X(*) - X COORDINATES FOR NODES OF AN ELEMENT
34 34 36 36 36 36 36 36 38
38 38 38 38 38 40 44

Y - Y COORDINATE MARIX
44

Y(*) - Y COORDINATES FOR NODES OF AN ELEMENT
34 34 36 36 36 36 36 36 38
38 38 38 38 38 40 44

Z - USER INTERACTION INDICATOR (0, MAX; 1,INTERMEDIATE; 2,FREE RUN; 3,DEMO)
1 174 184 184 230 230 230 246

ZC - CENTROID OF AN ELEMENT
34 34 34 34 34 38

1 FEB 85 Version JRC/DCD

END OF VAR. LIST

```
******************************
*                            *
*   POSTPROCESS.ELASTICITY   *
*                            *
*     TABLE OF VARIABLES     *
*                            *
******************************
```

*P1(*) - STEPPING FUNCTION (CW) AROUND ELEMENT
22 202

*P2(*) - STEPPING FUNCTION (CCW) AROUND AN ELEMENT
22 202

*R(*) - PRINTING SPACE BETWEEN VALUES
48 50 68 68 196

A$ - GENERAL INPUT VARIABLE
4 30 30 56 56 74 74 86 86
122 122 182 182 186 186

A2 - TWICE THE AREA OF AN ELEMENT
18 22 22 22 78 78 78

AN - INPUT ANGLE (DEGREES); INCLUDED ANGLE
60 60 62 62 62

AN$ - ANGLE
68 112 202

B - BOUNDARY NUMBER
20 26 36 38

B(*,*) - & BEEP COMMAND; DERIVATIVES OF SHAPE FUNCTIONS OF AN ELEMENT
4 22 22 22 22 22 22 22 26
194 216 220

B(*,*,*) - &BEEP (WITH REPETITION)
220 228

BC% - BRANCH CODE (1 ='<' BACKWARD; 2 ='>' FORWARD BRANCH; 3 = '.' A PREVIOUS VALUE
4 30 30 56 56 74 74 86 86
100 100 146 152 162 212 212
214 214 214 216 226 226 228

BW - BANDWIDTH OF STIFFNESS MATRIX
6 6 6 70 90 90 90 114 114
120 198

C% - FLAG FOR BRANCHING IN PROGRAM
18 34 76

C(*) - CLEAR ARRAY (& C(*))
52 52 58 70 88 104 106 138
192 194 216

C(*,*) - & CLEAR ARRAY CMD
46 130

C(*,*,*) - & CLEAR ARRAY (3 PARM)
114 128

C(*,*,*,*,*) - & CLEAR ARRAY (5 PARM)
70 92

C(*,*,*,*,*,*) - & CLEAR ARRAY CMD
28

C(*,*,*,*,*,*,*) - & CLEAR ARRAY (7 PARM)
26

C(*,*,*,*,*,*,*,*,*) - & CLEAR ARRAY CMD
38

C1 - GENERAL VARIABLE
6 8 78 78

C2 - GENERAL VARIABLE
8 8 8 78 78

CE - ELEMENT CONJUGATE STIFFNESS
70 78 92

CE(*,*) - ELEMENT CONJUGATE STIFFNESS
70 78 82 84

CR$(*) - STRING FOR CARRIAGE RETURN
198 198

CS - CONJUGATE STIFFNESS MATRIX
70 92

CS(*,*) - CONJUGATE STIFFNESS FOR NODE STRESS CALCULATIONS (E)
6 8 8 8 8 70 84 84 84 90
90 90

D - ELEMENT ELASTICITY MATRIX
12 26 36 38

D$ - CHR$(4) CONTROL D
1 150 150 150 154 154 154 160
172 172 172 182 186 218

D(*) - & DISPLAY
(&D(T)=TEXT,&D(G)=MIXED,&D(V)=FULL
GRAPHICS)
1

D(*,*) - ELEMENT MATERIAL PROPERTY
ELASTICITY MATRIX
12 12 14 14 16 16 16 26

D1$ - ROW DESCRIPTOR FOR SAVED DATA
150 154 154 170

D2$ - COLUMN DESCRIPTOR FOR SAVED
DATA
150 154 154 170

DC - COORDINATES OF DISPLACED NODES
130 136 138

DC(*,*) - COORDINATES OF DISPLACED
NODES
130 134

DD - DIMENSION OF ELEMENT ELASTICITY
MATRIX
8 14 14 14 14 16 16 16 20
20 22 22 22 26 26 26 28 28
28 28 28 28 36 38 42 48 48
50 50 52 60 60 70 70 70 78
84 88 90 92 98 98 198 200

DE - # NODES / ELEMENT
18 18 18 22 26 26 26 34 70
70 70 78 80 80 80 84 84 198
198 198 202 202 202

DE$ - DESCRIPTION OF DATA IN FNAMES
150 152 154 154 162 170

DF$(*) - COORDINATE (DIRECTION)
DESCRIPTOR
200 200 200 200 200 200

DI$(*) - GENERAL COORDINATE (DIRECTION)
DESCRIPTOR
48 50 98 108 110 122 122 200
200 200 200 200 200 202

DK - DIMENSION OF ELEMENT STIFFNESS
MATRIX
26 26 198

DR - DISKETTE DRIVE NUMBER
150 154 160 166 172 182 186
194 194 216

DR$ - DISKETTE DRIVE STRING
194

DR$(*) - DISKETTE DRIVE STRING
150 154 160 166 172 182 186
194 194 194

DT - DIFFERENCE BETWEEN ELEMENT
TEMPERATURE AND REFERENCE
TEMPERATURE
20 20

E0 - INITIAL THERMAL STRAIN
28 36 38

E0(*,*) - INITIAL THERMAL STRAIN
20 20 28

E1 - INTERMEDIATE VALUE IN STRAIN
CALCULATIONS
28 36 36 38

E1(*,*) - INTERMEDIATE VALUE IN STRAIN
CLCULATIONS
28 36

E2 - USED IN STRAIN CALCULATIONS;
ERROR FLAG
28 36 36 38 160 160 184

E2(*,*) - USED IN ELEMENT STRAIN
CALCULATIONS
28

EF - ERROR FLAG (1=ERROR OCCURRED;
0=NONE)
28 28 40 42 44 46 54 54 64
66 72 72 94 96 106 106 116
116 118 118 126 128 132 132
136 138 142 142 144 144 176
224 224

EL - ELEMENT # (GENERAL)
12 12 12 12 14 14 14 14 14
14 14 14 18 20 20 20 32 34
34 36 36 38 48 48 50 50 76
76 78 80 80 84 84 86

EN% - NODE # OF ELEMENT
142

EN%(*,*) - ELEMENT NODE #S (ELEMENT#,
LOCAL NODE #)
18 80 80 84 84 198 198

EO - EXIT OPTION (1=PROCEED TO NEXT
PROGRAM; 2=TO MAIN MENU;
3=REMAIN IN
SAME; 0=STOP)
208 214 214 214 214 214 216
218

ER - ERROR CODE
176 176 176 176 176 176 176
176 176 176 176 176 176 176
182 182 182 182

ER$ - ERROR CODE STRING DESCRIPTORS
192

ER$(*) - ERROR CODE STRING DESCRIPTORS
182 192 192 192 192 192 194
194

FI$ - 'FILEINFO.TXT' STRING
1 150 150 150 154 154 154 160
172 172 172

FR - FREE RUN FLAG (1=FREE RUN; 0=USER CONTROL)

4 4 4 4 30 56 74 86 86 100
106 122 124 152 154 156 162
170 208 208 216 226 226 226
226 226 226 228 228

GS - GLOBAL STIFFNESS MATRIX
114 116 128

GS(*,*) - GLOBAL STIFFNESS MATRIX
114 120 120 122

I - GENERAL INDEX
6 6 6 6 6 6 8 8 8 8 8 14
14 14 14 16 16 16 16 16 16
16 16 16 18 18 18 18 18 18
18 18 18 18 18 18 18 18 20
20 20 22 22 22 22 22 22 22
34 34 34 34 34 34 36 36 36
36 36 36 48 48 48 48 50 50
50 50 78 78 78 78 78 80 80
80 82 82 84 84 84 84 86 90
90 90 90 90 90 90 90 90 98
98 98 98 100 100 100 100 106
106 106 108 108 108 108 108
108 108 108 110 110 110 110
110 110 110 112 112 112 112
112 118 118 120 120 120 122
122 122 122 122 122 122 122
134 134 134 134 134 188 188
196 202 202 202 202 202 202
206 206 206 206 210 210 210
210

I(*,*,*,*) - & INPUT CMD
4 56 74 86 122 182 186

I(*,*,*,*,*) - & INPUT CMD
30

I(*,*,*,*,*,*) - & INPUT CMD
100 152 162 212 214 226

I1 - GENERAL SUBSCRIPT; STARTING SUBSCRIPT FOR A BOUNDARY
120 120

J - GENERAL SUBSCRIPT
8 8 8 8 8 8 14 14 14 16 16
16 16 22 22 22 22 22 22 22
22 22 22 24 48 48 48 50 50
50 68 68 68 68 68 68 68 68
78 78 78 78 80 80 80 80 80
80 80 82 82 84 84 84 86 90
90 90 90 98 98 98 98 98 98
98 98 98 110 110 110 110 110
110 112 112 112 112 120 120
120 120 122 134 134 134 134
134

K - GENERAL SUBSCRIPT
8 8 8 8 8 14 14 20 20 22
22 22 22 22 22 22 78 78 84
84 84 84 90 90 90 120 120
120 120 120 120 120 120 120
120

KI - GENERAL INDEX
84 84 84 84 84 84 84 84 84

KJ - GENERAL INDEX
84 84 84 84

KW$ - KEYWORD
164 164 164 172 174 174

L - GENERAL VARIABLE
8 8 8 8 8 84 84 84 84 84
90 90 90 90 90 90 90

L(*) - & LISTING CONTROL
1 48 50 68 68 80 86 98 98
108 108 110 110 112 112 122
124

L2 - LISTING FLAG
100 100 208 212

LO - LISTING OPTION
100 100 100 100 102 102 102
102 102 102 102 102 106 106
106 106 106 106 106 106 106
106 106 106 106 106

LO$(*) - LISTING OPTION TITLES
100 102 106 204 206 206 206
206

M - SUBSCRIPT
120 120 120

M(*) - & MATRIX CMD
12 20 34 34 36 36 36 78 230
230 230 230 230 230 230 230
230

M1 - SUBSCRIPT
120 120

MJ - SUBSCRIPT
6 6 6 6 6 8 90 90

MK - SUBSCRIPT
6 6 8 8 8

MO - MENU OPTIONS
52 52 52 52 52 58 58 58 62
62 62 62 66 68 104 106 106
208 212 212 212 212 212 226
228 228 228 228 228 228

MO(*) - TEMP ARRAY FOR OUTPUT
104 108 108

MO(*,*) - PRINCIPAL STRESSES AND ANGLE FROM X AXIS
102 110 112 112

MP - MATERIAL PROPERTY ARRAY
224

MP(*,*) - MATERIAL PROPERTY ARRAY
12 12 12 12 14 14 14 14 14
14 14 14 20 20 20 20 198 202
202 202 202 202

MU$(*) - MENU (2.05) OPTION STRING (GRID)
204 204 204 204 204 204 204
206 210 212 212 228

N - SUBSCRIPT; COMMONLY IN STEPPING FUNCTION
6 8 8 8 8 52 52 58 58 58
58 90 90

N$ - GENERAL STRING
28 40 44 54 64 72 94 106 116
118 126 132 136 142 144 166

N1 - GENERAL INDEX; NUMBER OF ROWS IN SAVED ARRAY
38 42 62 92 102 104 108 110
112 114 124 134 150 150 154
154 170 198 198

N2 - GENERAL INDEX; NUMBER OF COLUMNS IN SAVED ARRAY
38 42 62 92 102 104 110 110
114 124 134 150 154 154 170
198

NA$ - FILENAME
146 150 152 152 154 156 162
162 164 164 164 166 168 170
182 224

NC - NODE COORDINATES
144

NC(*,*) - COORDINATES OF NODES (ABSOLUTE OR SCREEN COORDINATES)
18 18 134 198

ND - NODE NUMBER
26 38

ND(*) - NODE NUMBERS FOR AN ELEMENT
18 18 18 18 18 26 34 34

NE - NUMBER OF ELEMENTS
28 28 34 34 38 42 52 58 62
70 76 76 198 198 198

NM - NEXT MENU DEFAULT
52 208 212

NN - NUMBER OF NODES
6 6 6 6 6 6 6 52 58 62 70
70 88 90 90 90 90 90 90 90
90 92 98 130 134 134 198 198 198

NO - GENERAL SUBSCRIPT
58 58 60 60 60 60 60 60 60
60 60 60 62 62 62 62 62 62
62 68 68 68 68 68

NP - NUMBER OF DATA POINTS; # D.O.F.
26 114 114 114 114 118 118
120 122 124 130 198

NS(*) - DEGREE OF FREEDOM FOR NODE AND DIRECTION
18 18

P% - PARAMETER IN LISTING SPEED ROUTINE
1 48 68 80 98 108 110 112
122

PD$ - PROBLEM DESCRIPTOR
172 174

PN$ - PROGRAM NAMES
216

PN$(*) - PROGRAM NAMES
216 216 216 216 218

PO - PRINT OR PLOT OPTION
30 30 38 56 56 58 62 74 74
80 84 86 86 86 92

PS - PRINCIPAL STRESSES; PLANE STRESS/ PLANE STRAIN INDICATOR
12 14 20 30 58 64

PS(*,*) - PRINCIPAL STRESSES AND THEIR ANGLES
58 62 62 62 62 62 62 68 68
68 68

R - RATIO OF MATERIAL PROPERTIES
12 12 12 14 14 14 16

R(*,*) - & RECALL ARRAY
28 54 72 106 116 118 132 142
144 224

RC - CENTROID OF ELEMENT (E)
18 18 18 18 18 22 22

RD - RADIANS/DEGREE CONVERSION
60 202

RE - REGION # OR RECORD #
26 26 38 42 52 52 62 70 70
70 92 92 92 102 102 114 114
116 116 124 130 130 134 140
140 142 142 150 150 154 154
154 198 198 222

RE(*,*) - INTERMEDIATE DATA ARRAY IN NODAL STRESSES
70 78 84

RH - RIGHT HAND SIDE MATRIX
70 92 114 126 128

RH(*) - RIGHT HAND SIDE OF MATRIX EQUATION
114 122 122 122

RH(*,*) - RIGHT HAND SIDE
8 8 8 70 84 84 90 90

RW - MATRIX ROW TYPE
106 108 110 112 120 120 120
120 120 120 120 120

RW$(*) - DESCRIPTOR OF MATRIX ROW TYPE,RW
108 110 112 204 204 206

S$(*) - OPTION LABELS
108 110 204 206 206 206 206
206

S(*,*) - & SAVE ARRAY CMD
40 44 64 94 126 136

S1 - STRESS VALUE
60 62

S2 - STRESS VALUE
60 62

SI - INTERMEDIATE STRESS MATRIX
28 36 38

SI%(*) - STATUS INDICATOR (0=INACTIVE; 1=ACTIVE)
26 32 32 52 70 92 102 114
116 130 140 142 150 154 154
198

SI(*,*) - INTERMEDIATE VALUES IN STRESS CALCULATION
28 36

SM - MEAN STRESS
60 60 60

SN - STRAINS
28 34 40 46

SN(*,*) - NODAL STRAINS
28 36 48 48

SO - OUTPUT OPTION
102 108 110

SR$ - 'SELECT A NUMBER' PROMPT
1 100 212 214 226

SS - STRESSES
28 34 44 46 52 54 70 70 72
92

SS(*,*) - STRESSES
28 36 50 50 52 60 60 60 60
60 60 60 60 60 60 60 70 78

ST - NODAL STRESS

SU - SUM IN MATRIX MULTIPLICATION
90 90 90 90 120 120 120 120SUM IN MATRIX MULTIPLICATION
90 90 90 90 120 120 120 120
120 122

T - &D(T) SETS TEXT DISPLAY
1

T$ - STRING DEFINING NODE OR ELEMENT
52

T$(*) - NODE/ELEMENT STRING
52 52 52 52 58 68

TE - TYPE OF ELEMENT (1=TWO-DIMENSIONAL; 2=AXISYMMETRIC)
12 14 14 20 22 172 198 198
200 200 202

TM - MAXIMUM SHEAR STRESS
60 60 60 62

U - NODE DISPLACEMENTS FOR AN ELEMENT
26 36

U(*,*) - NODE DISPLACEMENTS FOR AN ELEMENT
26 34 34

UN - UNIFORMITY OF MATERIAL PROPERTY FLAG (0=NOT, 1=UNIFORM)
32 34 202

UU - DISPLACEMENTS FOR ALL NODES
26 28 114 118 128 130 132

UU(*) - SOLUTION NODE DISPLACEMENTS FOR ALL NODES
26 30 34 34 114 120 120 122
130 134

W - INDEX
118 120 120 120 122

X - X COORDINATE MATRIX; DUMMY ARGUMENT
26 38 196 196

X(*) - X COORDINATES FOR NODES OF AN ELEMENT
18 18 18 18 18 18 18 18 22
22 22 22 22 22 26

Y - Y COORDINATE MATRIX
26 38

Y(*) - Y COORDINATES FOR NODES OF AN ELEMENT
18 18 18 18 18 18 18 18 22
22 22 22 22 22 26

Z - USER INTERACTION INDICATOR (0=MAX; 1=INTERMEDIATE; 2=FREE RUN; 3=DEMO)
1 118 120 120 120 152 162 162
208 208 208 208 226

ZC - CENTROID OF AN ELEMENT
18 18 18 18 18 22

1 FEB 85 Version JRC/DCD

END OF VAR. LIST

```
*******************************
*                             *
*       PLOT.ELASTICITY       *
*                             *
*     TABLE OF VARIABLES      *
*                             *
*******************************
```

*AO(*) - TEST FOR LINE ENTIRELY OFF SCREEN (1=OFF; 2=NOT)
26 28 30 298

*C(*) - STEPPING FUNCTION AROUND CONTOUR LINES
16 296

*LE(*) - DEFINES LENGTH OF LINE
36 58 102 296

*M(*) - STEPPING FUNCTION AROUND BOUNDARIES
58 58 126 302

*P2(*) - STEPPING FUNCTION (CCW) AROUND AN ELEMENT
38 94 302

*PX(*) - CONVERTS X ABSOLUTE COORDINATE TO SCREEN COORDINATE
60 62 100 138 140 158 158 306

*PY(*) - CONVERTS Y ABSOLUTE COORDINATE TO SCREEN COORDINATE
60 62 100 140 158 158 306

*S(*) - STEPPING FUNCTION AROUND A MATRIX OF NP VALUES (UP IF A=21; DOWN IF A=8)
12 302

*UX(*) - CONVERTS X SCREEN COORDINATE TO ABSOLUTE COORDINATE
136 138 306

*VY(*) - CONVERTS Y SCREEN COORDINATE TO ABSOLUTE COORDINATE
136 138 306

*W(*) - TEST FOR POINT OFF SCREEN (1=OFF;0=NOT)
40 56 82 302

*WM(*) - STEPPING FUNCTION AROUND ZONES BEING SELECTED FOR PLOTTING
150 150 312

*ZX(*) - DEFINITION OF X COORDINATE FOR DOT IN ZONE SELECTION
132 132 150 150 312

*ZY(*) - DEFINITION OF Y COORDINATE FOR DOT IN ZONE SELECTION
132 132 150 150 312

A - GENERAL VARIABLE; FREQUENTLY ASCII VALUE OF INPUT CHARACTER
12 12 12 12 12 12 12 16 16
16 18 18 18 18 18 18 18 40
50 56 56 58 58 58 58 68 68
68 72 72 72 72 72 72 80 82
82 82 108 108 108 112 112 112
112 126 142 142 142 150 226
226 226 226 226 242 242 242
296 296 302 302 302 302 302
312 312 312 312 340 356 356
356 358 358 358 360 360 360
360 360 364 366 368 368 368
368 368

A$ - GENERAL INPUT VARIABLE
4 90 96 96 118 118 142 144
180 180 184 184 192 192 194
194 204 206 206 206 206 210
210 212 214 216 216 218 218
220 220 222 222 228 228 228
230 230 230 230 278 278 282
282 330 332 334

A% - ASCII CODE OF INPUT CHARACTER; LOGICAL VARIABLE
6 6 6 6 6 6 6 44 44 44 44
48 48 48 48 48 48 94 96 160
160 186 186 196 208 212 216

A%(*) - STATUS INDICATOR FOR LABEL
12 18 18 18 48 196 198 206
212

AN - INPUT ANGLE (DEGREES); INCLUDED ANGLE
82 82 82 340 340

AN(*) - ANGLE OF PRINCIPAL STRESS CCW FROM X-DIRECTION
82 340 340

B - BOUNDARY NUMBER
56 58 58 58 58 58 58 126 126
126 126 130 302 302 302 302
302 302 302 302 302

B(*,*) - & BEEP COMMAND
4 6 12 16 292 320 350

B1 - GENERAL VARIABLE
126 126 126 128

B2 - GENERAL VARIABLE
126 126 126 128

BC - EQUIVALENT BOUNDARY CONDITION MATRIX
106 108 324

BC% - BRANCH CODE (1 ='<' BACKWARD; 2 ='>' FORWARD BRANCH; 3 = ':' A PREVIOUS VALUE
4 120 120 120 120 120 134 142
142 144 144 144 170 172 176
176 178 178 178 180 182 182
196 196 210 210 210 212 218
218 218 218 226 226 226 228
228 248 258 258 304 316 318
318 320 320 324 326 326 330
330 332 332 336 336 338 338
340 342 342 344 344 346 346
348 350 352 356 360 360

BC(*,*) - EQUIVALENT BOUNDARY CONDITIONS
56 56 60 60 60 60 60 60 68
68 68 68 72 72 106 108 108 108

BN% - BOUNDARY NODES
246

BN%(*,*) - BOUNDARY NODES ARRAY WITH NODES IN ORDER AROUND BOUNDARIES
56 58 58 126 126 304 304 306

BR% - BEGINNING ROW NUMBER IN &W()
96 364 364 364 364 364 368
368 368 368 368

C - GENERAL
38 38 38 40 40 40 40 48 48
48 48

C% - STEP SIZE IN MOVING LABELS ON PLOTS; GENERAL VARIABLE (E)
6 44 44 44 44 44 44 44 44
48 48 48 48 48 48 48 48 86 86

C(*) - WEIGHTING FACTOR FOR FINDING CENTROIDS OF CLIPPED ELEMENT; CLEAR ARRAY (& C(*)) (E)
26 26 28 30 38 38 38 96 106
110 114 144 292 294 304 324
326 330 340 342 350

C(*,*) - & CLEAR ARRAY CMD
94 212 216 340

C(*,*,*) - & CLEAR ARRAY (3 PARM)
316 340

C(*,*,*,*) - & CLEAR CMD
196 208 240

C1 - GENERAL VARIABLE
72 74 74 76 76 76 76

C2 - GENERAL VARIABLE
72 74 74 76 76 76 76

CC - MATRIX OF ELEMENT CENTROIDS
234 240 246 246 246

CC(*,*) - COORDINATES OF ELEMENT CENTROIDS
240 246 246 246 246 246

CF - FLAG INDICATING CONTOUR POINT SAVED
48 50 194 210

CL - CONTOUR LINE INDEX
14 14 14 14 16 16 16 94 94
94 94 94 96 96 216 216 216

CL(*,*) - CONTOUR LINE END POINT COORDINATES
20 20 20 94

CO - CONTOUR LABEL OPTION (1=LETTER; 2=NUMBER)
16 210 210 210 210 214 216

CO(*) - PARAMETER FOR PATTERNS IN &FILL
86 296

CP - CONTOUR LINE PLOT OPTION (1=CONTOURS ONLY; 2=WITH96ESH)
186 186 338 338 338

CV - CONTOUR VALUE; MATRIX OF COORDINATES OF CONTOUR LABELS
16 16 20 94 94 94 94 96 216
216

CV(*,*) - COORDINATES OF POINTS WHERE CONTOURS ARE LABELED
14 14 14 16 16 16 16 16 16

28 28 28 28 28 28 30 30 30
30 30 30 30 30 30 30 30 30
30 32 32 32

EO - EXIT OPTION (1=PROCEED TO NEXT PROGRAM; 2=TO MAIN MENU; 3=REMAIN IN SAME; 0=STOP)
318 318 318 318 318

ER - ERROR CODE
272 272 272 272 272 272 272
272 272 272 272 272 272 272
278 278 278 278 278 278 278

ER$ - ERROR CODE STRING DESCRIPTORS
294

ER$(*) - ERROR CODE STRING DESCRIPTORS
278 294 294 294 294 294 294 294

ER% - ENDING ROW NUMBER RETURNED FROM A VARIABLE SEARCH
96 96 364 364 364 364 368 368
368 368

EX(*) - X SCREEN COORDINATE OF LINE END
14 14 20 22 22 26 26 28 28
28 28 30 30 30 30 30 30 34
34 36 36 38 38 38 122 122
122 126 126 128 128 128 128

EY(*) - Y SCREEN COORDINATE OF LINE END
14 14 20 22 22 26 26 28 28
28 28 28 28 30 30 30 30 34
34 36 36 38 38 38 122 122
122 126 126 128 128 128 128

F$ - GENERAL STRING
88

F(*,*,*,*) - AMPERSAND FILL COMMAND
86

F1 - TEMP VARIABLE
60 60 60 62 62 64 64

F2 - TEMP VARIABLE
60 60 60 62 62 64 64

FH - MAXIMUM FORCE MAGNITUDE
108 108 108 108 108 108 108
192 192 192

FI$ - 'FILEINFO.TXT' STRING
252 252 252 268 268 268 284

FM - MAXIMUM FORCE MAGNITUDE
60 60 62 62 192

FR - FREE RUN FLAG (1=FREE RUN; 0=USER CONTROL)
20 20

G - PARAMETER IN &D(G) SETS MIXED TEXT AND GRAPHICS
44 52 138 208 216 232 358

G$ - GENERAL INPUT CHARACTER
6 6 12 12 16 16 46 46 46 46
136 136 138 138 140 142 150
150 150 150 150 150 150 152

GC - GRADIENT COMPONENT OPTION
214

GC$(*) - GRADIENT COMPONENT DESCRIPTORS
214

GD - GRADIENT DIRECTION INDICATOR
186 188 344 344 344 344

GL - MESH TYPE (1=DASHED ORIG,SOLID DISPLACEMENT; 2=SOLID ORIGINAL,DASHED DISPLACEMENT)
186 188 346 346 346 346

H% - HORIZONTAL INDEX
180 180 180

HE$ - GENERAL HEADING
316 316 316

I - GENERAL INDEX
26 28 30 36 58 80 80 80 80
80 80 82 82 82 82 84 90 90
90 90 94 94 94 96 100 100
100 100 100 100 102 104 108
108 108 108 108 116 116 116
116 122 122 122 122 122 122
122 122 124 132 132 132 144
144 144 146 148 148 148 148
148 148 150 150 150 150 150
150 150 150 150 150 152 152
152 152 154 154 154 154 154
154 156 156 156 158 158 158
158 158 158 158 158 198 198
198 198 198 198 206 206 206
206 206 206 206 212 212 212
214 214 214 214 232 232 232
232 232 232 234 234 234 234
234 234 242 242 242 242 242
242 246 246 246 246 246 246
246 246 246 246 246 286 286
296 298 302 302 302 306 306
306 306 312 312 312 312 312
312 312 328 328 328 328 332
332 332 332 334 334 334 340
340 340 340 340 340 354 354
354 354 364 364 364 364 364
364 368 368 368 368 368 368

I% - STATUS INDICATORS FOR LABELS
196 208 212 216

I%(*) - STATUS INDICATORS FOR LABELS
12 18 18 48 48 48 196 198
206 206 212 212 216 364 368

MF - MULTIPLICATION FACTOR FOR PLOT SIZE
56 56 56 80 80 80 142 142
142 144 144 144 144 146 148
148 150 152 152 154 154 154
154 156 156 156 156 156 156
172 178 178 178 178 178 180
180 180 182 182 190 190 312
312 312 312 312

MO - MENU OPTIONS
80 100 186 186 186 186 186
186 186 186 186 186 188 190
190 192 192 192 192 192 192
210 214 228 274 274 316 316
316 316 316 316 316 324 330
330 334 334 334 334 336 340
340

MU$(*) - MENU (2.05) OPTION STRING (GRID)
228 296 308 308 308 308 308
308 308 308 314 314 314 314
314 314 316 316

N - SUBSCRIPT; COMMONLY IN STEPPING FUNCTION
12 12 12 12 12 12 12 12 12
12 16 16 16 16 16 16 16 16
16 18 18 18 18 18 48 48 48
48 48 48 50 50 56 56 58 58
58 70 78 126 126 126 130 194
196 196 196 196 196 196 204
204 204 204 204 204 212 216
216 216 216 216 216 220 252
296 296 302 302 302 302 302
302 364 364 364 364

N$ - GENERAL STRING
108 112 112 116 240 242 244
246 262

N% - GENERAL VARIABLE; NUMBER OF VARIABLES FOUND IN A SEARCH
86 94196 96 114 114 122 122
122 122 122 168 354 354 354
354 356 364 364 364 364 364
368 368 368 368

N(*,*,*,*,*) - AMPERSAND NAMES COMMAND; FIND CATALOG ENTRIES ENDING WITH '.PIC'
354

N1 - GENERAL INDEX; NUMBER OF ROWS IN SAVED ARRAY
20 20 22 22 58 58 58 94 94
100 100 106 110 114 126 126
252 264 304 304 304 304

N2 - GENERAL INDEX; NUMBER OF COLUMNS IN SAVED ARRAY
20 22 22 58 58 58 94 94 100
100 100 114 252 264 304

NA$ - FILENAME
248 252 258 258 260 260 260
262 266 278

NB - # OF BOUNDARY NODES IN BN%
56 304 304

NC - NODE COORDINATES
232 240 368 368 368

NC(*,*) - COORDINATES OF NODES (ABSOLUTE OR SCREEN COORDINATES)
20 20 20 22 22 22 22 38 38
38 38 56 56 64 64 68 68 70
70 74 74 76 76 80 80 100 100
122 122 126 126 126 126 232
232 232 232 240 242 368 368
368

ND - NODE #
36 36 36 36 72 72 74 110 112
112 112 316 342

ND% - ELEMENT NODE #S
240

ND%(*,*) - ELEMENT NODE #S
38 38 94 94 246 304

ND(*) - NODE NUMBERS FOR AN ELEMENT
100 100 110 112 112 112 112
122 122 128 128 128 128

NE - NUMBER OF ELEMENTS
86 86 94 196 234 240 240 246
302 302 302 302 304 304 364
366

NI - NUMBER OF INPUT VALUES
86 90 94 94 94 96 96 118 118
118 120 214

NL - # UNIQUE LINES
122 304 304

NM - NEXT MENU DEFAULT
314 316

NN - NUMBER OF NODES IF USER MODIFIED WITHOUT GRAPHICS
80 100 196 232 240 240 242
242 242 242 242 242 304 304
368

NO - GENERAL SUBSCRIPT
12 56 56 56 56 60 60 62 62
64 64 68 68 70 70 74 74 76
76 196 198 204 204 206 364
368

NP - NUMBER OF DEGREES OF FREEDOM
106 106 108 110 110 112 112
112 304

NR - # ROWS IN A MATRIX
80 114 114 116 340 340 340
340 340

NS - # SEPARATE BOUNDARIES
126 304 304 306

NV(*) - NORMAL VECTOR FOR AN ELEMENT SIDE
58 58 60 60 62 62 64 64 68
68 68 68 72 72 304

O% - PARAMETER IN &FILL ROUTINE
86 86

O(*,*,*,*) - & ORDER CMD (SORT)
112 242 242 242 340 364 366
368

OP(*) - FILL OPTION
86 296

P - INDEX IN CONTOUR PLOTTING
16 20 20 20 20 20 20 94 96
98 98 212 212 212 216 296 296
296 296 354

P% - PARAMETER IN LISTING SPEED ROUTINE
1

P2 - TWO*PI
72

PB - Y COORDINATE OF PICTURE BOTTOM FOR ZONE PLOT
154 154 156 156 156 158 158
158

PC - INDICATOR OF STRESS COMPONENT TO PLOT
330 330 330 330 330 330 330
332 332 332 332 332 340 340
340 340 340

PC$(*) - MAX/MIN/SHEAR STRESS STRING
296 310 312 312 332 332

PD - INDICATOR OF METHOD TO PLOT NODE DISPLACEMENTS (1=DISPLACED GRID;
2=VECTORS
100 186 186 188 192 342 342
342 342 344

PD$ - PROBLEM DESCRIPTOR
268 270

PF - FLAG TO BRANCH IN PLOTTING CONTOUR LINES
14 14 94

PF$ - PICTURE FILENAME ARRAY
350 354

PF$(*) - PICTURE FILENAME ARRAY
350 354 354 356 358 358

PL - X COORDINATE OF PICTURE LEFT FOR ZONE PLOT
154 154 156 156 156 158 158
158

PR - X COORDINATE OF PICTURE RIGHT FOR ZONE PLOT
156 156 158 158 158

PT - Y COORDINATE OF PICTURE TOP FOR ZONE PLOT
156 156 158 158 158

PV$(*) - PLOT VARIABLE DESCRIPTOR FOR LABELING NODES OR ELEMENTS
194 194 194 200 206 206 296
310 310

PX - DUMMY VARIABLE
306 306

PY - DUMMY VARIABLE
306 306

Q$ - QUIT/LEFT ARROW/RIGHT ARROW PROMPT
146 204 212 284

R(*,*) - & RECALL ARRAY
108 112 112 116 240 242 244
246

RA - RADIUS OF CIRCLE IN SYMBOL FOR B.C.
72 74 74 74 74 74 76 76 76
76

RD - RADIANS/DEGREE CONVERSION
82 298

RE - REGION # OR RECORD #
106 106 110 110 114 238 238
240 240 242 242 244 244 252
252 304 304 304 304 324 326
330 340

RS - RIGHT COORDINATE OF SCREEN
172 306

S$ - PROMPT "<S>TO SELECT"
1 134 146 202 220

S0 - SCALE FACTOR FOR FITTING WHOLE PLOT ON SCREEN
172 178 236 236 236 236 236

S1 - VALUE AT A NODE
20 20 20 94 94 94

S2 - VALUE AT A NODE
20 20 94 94 94

SB - SCREEN BOTTOM Y COORDINATE
26 28 30 92 92 92 98 98 136
154 154 154 156 156 156 180
180 224 224 232 236 302 306
306 306 364 368

SC - SCALE
104 122 128 172 178 180 180
180 180 232 232 232 234 236
236 236 236 236 236 306 306
306 306

SF - SCALE FACTOR
56 70 70 72 74 74 74 74 74
76 76 76 76

SF% - SECTORS FREE ON DATA DISKETTE
354 354

SI%(*) - STATUS INDICATOR FOR DATA FILES (0=INVALID; 1=VALID)
106 110 114 238 240 242 244
252 296 324 326 330

SL - SCREEN LEFT X COORDINATE
26 28 30 92 92 92 98 98 136
154 154 154 156 156 156 180
180 224 224 232 236 302 306
306 306 364 368

SP - SLOPE OF A LINE
28 28 30 30

SR - SCREEN RIGHT X COORDINATE
26 28 30 92 92 98 98 140 154
154 156 156 172 180 180 224
224 236 302 306 306 364 368

SR$ - 'SELECT A NUMBER' PROMPT
1 144 196 210 226 316 318 326
330 332 336 338 342 344 346 356 360

ST - NODAL STRESSES (E)
26 28 30 92 92 98 98 114 116
138 140 154 154 156 156 172
180 180 224 224 236 302 306
306 316 326 330 340 340 364
368

ST(*,*) - NODAL STRESSES
114 116 340 340 340 340

SX - X-COMPONENT OF LINE SEGMENT
36 36 36

SY - Y-COMPONENT OF LINE SEGMENT
36 36 36

SZ - SIZE INDICATOR FOR PLOTTING B.C. SYMBOLS
186

T - &D(T) SETS TEXT DISPLAY
1 54 284 314 324 340 348 362

TE - TYPE OF ELEMENT (1=TWO-DIMENSIONAL; 2=AXISYMMETRIC)
268 296 300 300 300

TP - INDICATOR OF PLOT TYPE
118 186 186 190 192 210 334
334 334 336 336 336 336 336
336 336

TP$(*) - DESCRIPTOR OF PLOT TYPES: SHADED, CONTOURS, VECTORS, ETC.
296 310 310 310 334

TS - INDICATOR OF TYPE OF STRESS
88 326 326 326 326 328 328
328 328 328 334 334 340 340

TS$(*) - TYPE OF STRESS DESCRIPTORS: COORDINATE DIRECTION, PRINCIPAL
88 296 310 310 326 326 328

TT - TOP COORDINATE OF SCREEN
172 306

TV - TEMPORARY VALUE
28 28 28 30 30 30

U$ - UNDERLINE STRING
1 294

UH - HIGH VALUE OF X COORDINATE IN SPECIFIC PLOT
138 140 140 172 180 236 306

UL - LOW VALUE OF X COORDINATE IN SPECIFIC PLOT
140 172 180 232 236 306 306
306

UX - DUMMY PARAMETER
306 306

V - PARAMETER IN &D(V) TO VIEW FULL GRAPHICS
44 134 142 160 196 204 206
212 214 216 220 224 234

V% - VERTICAL INDEX
180 180 180 180

V0 - MINIMUM VALUE
16 80 80 80 86 86 86 90 90
94 118 118 120 120 120 120
120 120 120 214 216 340

V1 - MAXIMUM VALUE
80 80 80 86 86 118 118 120
120 120 120 120 120 120 340

VA - GENERAL VARIABLE
86 86 86 86 316 340 340

VA(*) - VARIABLE BEING PLOTTED
82 82 82 86 94 94 340 340

VH - HIGH VALUE OF Y COORDINATE IN SPECIFIC PLOT
140 140 172 180 236 306

VL - LOW VALUE OF Y COORDINATE IN SPECIFIC PLOT
140 172 180 232 236 306 306 306

VY - DUMMY PARAMETER
306 306

W(*,*,*,*,*,*) - AMPERSAND WHICH COMMAND; SELECTS VALUES WITHIN SPECIFIED RANGE
96 364 364 368 368

WR$ - 'WHEN READY' PROMPT
1 4 204 212

X - GENERAL X COORDINATE; NODE COORDINATES
8 8 8 8 10 10 10 10 232 240 242 242 242 242

X(*,*) - NODE COORDINATES
58 58 58 58 60 60 62 62 100 100 240 242 242 242 242 242 246 246

X1 - GENERAL X COORDINATE; START OF LINE
36 36 36 36 58 58 60 60 60 62 64 64 64 66 68 70 70 70 70 74 74 74 74 76 76 76 100 100 100 100 102 102 102 104 104 136 140 158 296

X2 - GENERAL X COORDINATE; END OF LINE
36 36 58 58 62 62 62 62 64 64 64 66 66 66 66 66 70 70 70 100 102 102 102 102 102 102 104 104 104 104 104 104 138 140 158 296

XA - X COORDINATE OFFSET FOR LABEL ON PLOTS
8 8 8 10 10 96 156 156 160 164 166 168 196 204 212 216 220 232 232 234

XC - GENERAL X COORDINATE; CENTROID COORDINATE
38 38 38 40 40 56 80 80 82 82 82 86 136 138 302 302

XH - MAX X COORDINATE
172 188 192 242 306

XL - MIN X COORDINATE
172 180 180 188 192 196 208 242 306

XL(*) - X COORDINATES OF A LIST OF POINTS
12 12 12 48 196 198 206 364 368

XP - GENERAL SCALED X COORDINATE
8 8 8 8 8 10 10 10 10 12 16 44 44 44 44 44 44 46 48 48 48 48 48 50 98 98 98 98 98 98 136 136 136 138 138 138 156 156 160 164 164 166 168 198 206 216 220 220

XP(*) - X COORDINATES FOR NODES OF AN ELEMENT
304

XY - COORDINATES OF POINTS
340

Y - GENERAL Y COORDINATE
8 8 8 10 10 10

Y1 - GENERAL Y COORDINATE; START OF LINE
36 36 36 36 58 58 60 60 60 62 64 64 64 66 68 70 70 70 70 74 74 74 74 76 76 76 100 100 100 100 102 102 102 104 104 136 140 158 296

Y2 - GENERAL Y COORDINATES; END OF LINE
36 36 58 58 62 62 62 62 64 64 64 66 66 66 66 66 70 70 70 100 102 102 102 102 102 102 104 104 104 104 104 104 138 140 158 296

YA - Y COORDINATE OFFSET FOR LABEL ON PLOTS
8 8 10 96 156 156 160 164 164 166 168 196 204 212 216 220 232 232 234

YC - GENERAL Y COORDINATE; CENTROID COORDINATES
38 38 38 40 40 56 80 80 82 82 82 86 136 138 302 302

YH - MAX Y COORDINATE
172 188 192 242 306

YL - MINIMUM Y COORDINATE
172 180 180 188 192 196 208 242 306

YL(*) - Y COORDINATES OF A LIST OF POINTS
12 12 12 48 196 198 206 364 368

YP - GENERAL SCALED Y COORDINATE
8 8 10 12 16 44 44 44 44 44

44 44 46 48 48 48 48 48 50
98 98 98 98 98 98 136 136
136 138 138 138 156 156 160
164 164 166 168 198 206 216
220 220

YP(*) - Y COORDINATES FOR NODES OF AN ELEMENT
304

Z - USER INTERACTION INDICATOR (0=MAX; 1=INTERMEDIATE; 2=FREE RUN; 3=DEMO)
258 258 284

Z% - ZONE SELECTION INDICATOR MATRIX
144

Z%(*,*) - INDICATOR OF SELECTION STATUS FOR ZONES TO BE PLOTTED
132 144 144 150 150 150 150
152 152 178

ZI - INDEX FOR ZONES
178 178 180 182 182 182

ZJ - INDEX FOR ZONES
178 178 180 182 182 182

ZN - ZONE NUMBER
42 156 156 156 156 162 162
172 180 180 180 224

ZS - OPTION FOR SELECTING ZONES TO BE PLOTTED (ALL, SELECT, OMIT)
144 144 144 144 146 152

ZS$(*) - DESCRIPTORS FOR ZS
146 296 300 300

ZT - TYPE OF ZONE TO BE PLOTTED (1=SINGLE; 2=MULTIPLE EQUAL-SIZED)
14 42 42 42 42 86 88 162 172
172 176 176 176 178 178 178
178 178 178 178 178 184 224

1 FEB 85 Version JRC/DCD

END OF VAR. LIST

```
******************************
*                            *
*  DISKETTE PREP.ELASTICITY  *
*                            *
*     TABLE OF VARIABLES     *
*                            *
******************************
```

A - GENERAL VARIABLE
34 34 34 34

A$ - GENERAL INPUT VARIABLE
4 6 6 6 6 32 32 32 36 48

B(*,*) - & BEEP COMMAND
32 48

B(*,*,*) - & BEEP (WITH REPETITION)
32

BC% - BRANCH CODE (1 ='<' BACKWARD; 2 ='>' FORWARD BRANCH; 3 = ':' A PREVIOUS VALUE
8 8 8 8

C(*,*,*,*) - & CLEAR CMD
8

D$ - CHR$(4) CONTROL D
1 6 8 28 28 28 28 30 30 32
38

D(*) - & DISPLAY (&D(T)=TEXT,&D(G)=MIXED,&D(V)=FULL GRAPHICS)
1

D1$ - ROW DESCRIPTOR FOR SAVED DATA
8

D1$(*)
8 28 30

D2$ - COLUMN DESCRIPTOR FOR SAVED DATA
8

D2$(*)
8 28 30

DE$ - DESCRIPTION OF DATA IN FILEINFO.TXT
8

DE$(*) - DESCRIPTOR STRING FOR DATA FILES
8 28 30

DR - DISKETTE DRIVE NUMBER
4 4 6 8 28 32 36

DR$(*) - DISKETTE DRIVE STRING
4 4 6 8 28 32

DS$ - PROBLEM DESCRIPTION
8 28

ER - ERROR CODE
42 42 42 42 42

FI$ - 'FILEINFO.TXT' STRING
1 28 28 28 28 28 30 30 32

I - GENERAL INDEX
28 28 28 28 28 28 40 40

I(*,*) - & INPUT CMD
4 32 36 48

I(*,*,*,*) - & INPUT CMD
6 6 8 32

I(*,*,*,*,*) - & INPUT CMD
34

I(*,*,*,*,*,*) - & INPUT CMD
8

KW$ - KEYWORD
8 28 30

N - GENERAL VARIABLE=1
28 28 30 30

NA$ - FILEINFO.TXT
8

NA$(*) - FILEINFO.TXT
8 28 30

RE - RECORD #
30 30 30 30 30 30 30

S - GENERAL VARIABLE=0
28 30

T - &D(T) SETS TEXT DISPLAY
1

WR$ - 'WHEN READY' PROMPT
1 32 36 48

1 FEB 85 Version JRC/DCD

END OF VAR. LIST

```
*****************************
*                           *
*     TEXT.ELASTICITY       *
*                           *
*    TABLE OF VARIABLES     *
*                           *
*****************************
```

A$ - GENERAL INPUT VARIABLE
502 508

B(*,*) - & BEEP COMMAND
6 508

BC - EQUIVALENT BOUNDARY CONDITION MATRIX
18 18

BC% - BRANCH CODE (1 ='<' BACKWARD; 2 ='>' FORWARD BRANCH; 3 =':' A PREVIOUS VALUE
502 502

BC(*,*) - EQUIVALENT BOUNDARY CONDITIONS
18

BN% - BOUNDARY NODES
12 12

BN%(*,*) - BOUNDARY NODES ARRAY WITH NODES IN ORDER AROUND BOUNDARIES
12

C(*) - CLEAR ARRAY (&C(*))
6 8 10 10 12 12 14 18

D$ - CHR$(4) CONTROL D
8 8 8 500 500 500

D(*) - & DISPLAY (&D(T)=TEXT,&D(G)=MIXED,&D(V)=FULL GRAPHICS)
2

D1$ - ROW DESCRIPTOR FOR SAVED DATA
500 506

D2$ - COLUMN DESCRIPTOR FOR SAVED DATA
500 506

DE - # NODES / ELEMENT
10 10

DE$ - DESCRIPTION OF DATA IN FILEINFO.TXT
500 502 506

DR - DISKETTE DRIVE NUMBER
6 8 500 500 504

DR$ - DISKETTE DRIVE STRING
6

DR$(*) - DISKETTE DRIVE STRING
6 6 6 8 500 500 504

FI$ - 'FILEINFO.TXT' STRING
6 8 8 8 500 500 500

I - GENERAL INDEX
2 2

I(*,*) - & INPUT CMD
508

I(*,*,*,*,*,*) - & INPUT CMD
502

KW$ - KEYWORD
8 8 502 502 502

LN% - LINE NODE NUMBER ARRAY
12 12

LN%(*,*) - LINE NODE NUMBER ARRAY
12

MP - MATERIAL PROPERTY ARRAY
14 14

MP(*,*) - MATERIAL PROPERTY ARRAY
14

N$ - GENERAL STRING
10 10 12 12 14 18 500 504

N1 - GENERAL INDEX; NUMBER OF ROWS IN SAVED ARRAY
10 10 12 12 18 500 506

N2 - GENERAL INDEX; NUMBER OF COLUMNS IN SAVED ARRAY
10 14 500 506

NA$ - FILENAME
500 500 502 502 502 502 504
510

NB - # OF BOUNDARY NODES IN BN%
12 12

ND% - ELEMENT NODE #S
10 10

ND%(*,*) - ELEMENT NODE #S
10

NE - NUMBER OF ELEMENTS
10 10 14

NL - NUMBER OF UNIQUE LINES
12 12

NN - NUMBER OF NODES
10 10

NP - NUMBER OF DEGREES OF FREEDOM
18 18

PD$ - PROBLEM DESCRIPTOR
8 8

R(*,*) - & RECALL ARRAY
10 10 12 12 14 18

RE - REGION # OR RECORD #
10 10 12 12 14 18 500 500
500

SI% - STATUS INDICATOR
8

SI%(*) - STATUS INDICATOR (0=INACTIVE; 1=ACTIVE)
8 500 500

T - &D(T) SETS TEXT DISPLAY
2

TE - TYPE OF ELEMENT (1=TWO-DIMENSIONAL; 2=AXISYMMETRIC)
8

XY - COORDINATES OF POINTS
10 10

XY(*,*) - COORDINATES OF POINTS
10

1 FEB 85 Version JRC/DCD

END OF VAR. LIST

3.4.2 Line Cross-References

```
******************************
*                            *
*      HELLO.ELASTICITY      *
*                            *
*    TABLE OF LN# XREFS      *
*                            *
******************************
```

```
     8
4

     10
8 20

     12
14

     14
12 12 16

     36
38

     38
34
```

1 FEB 85 Version JRC/DCD

END OF LINE LIST

```
******************************
*                            *
*    GEOMETRY.ELASTICITY     *
*                            *
*    TABLE OF LN# XREFS      *
*                            *
******************************
```

See the identical GEOMETRY.HEAT table in Chapter 2, page 249.

```
******************************
*                            *
*      GRID.ELASTICITY       *
*                            *
*    TABLE OF LN# XREFS      *
*                            *
******************************
```

See the identical GRID.HEAT table in Chapter 2, page 252.

```
******************************
*                            *
*  PREPROCESS.ELASTICITY     *
*                            *
*    TABLE OF LN# XREFS      *
*                            *
******************************
```

```
     4
6 102 126 192 224 282 290 298
316 324 326 330

     6
108 116 132 162 232 272 286 320

     8
34 46 52 330

     10
10 20 22 24 28 30

     12
20 22 24 26 32

     14
12

     16
20 22 24 26 32

     18
12 20 22 24 24 28 30 30 44 62
112 132 132 158 180 180 182 188
190 200 210 212 216 274 274 278
280 290 290 292 300

     20
152

     22
152

     26
26

     28
150

     32
32

     34
164

     40
164

     42
164

     44
224
```

46
14 16 64

48
330

50
46

52
46

56
96 132 312 326 328

60
96 102 120

62
62 64 328

64
132 326

66
60 68 102 126 126

68
120

70
80 274

72
114

74
74 74 106

76
80 84 84 88

84
80

86
88 106

96
88

98
96 98 98 98 100 100

100
98

102
98 98 100

104
84 104 104 108 126

108
74 80 84 88 94 104 104

110
72

112
70 116

116
114

118
96 120

120
96

122
120 122 122 122 124 124

124
122

126
122 122 124

128
156 290

130
128 132 134

134
134 138 138 138 142 142 146 146
148 150 152 152 152 154

138
138 142 146

142
140 142

146
140 140 146 148

148
144 148 152

150
148

152
152

154
134 134 154

158
156

160
162

162
160

168
168

170
168

172
250

176
248

178
180 182 182 188

180
180

182
182

186
188 190 190

188
188

190
190

192
178 180 186 188 270 284 318

194
192 218

196
158 166 178 180 186 188 220 270
274 280 284 318

198
160 168 198 276

200
162 170 278

202
158 178 238 270 280 318

204
204

206
180 182 188 190 206 272 286 320

208
198 198 198 206

210
198 206

212
180 182 188 190 272 286 320

214
238

216
170 278

218
158 166 178 270 274 284 318

220
218 224 224 224

222
218 226

224
218

226
202

228
1 258

238
234

256
2 258 258 262

260
256 258 258 266

270
274 292 292

272
272

274
258

276
276

278
276

280
258

282
280

284
290 292 292

286
286

```
    288
290  326

    290
280

    292
258

    294
292

    300
292

    304
302

    308
308

    310
308

    312
304

    314
292

    318
292  292

    320
320

    322
292

    326
292

    328
292

    332
274

1 FEB 85 Version JRC/DCD

END OF LINE LIST
```

```
*******************************
*                             *
*      SOLVE.ELASTICITY       *
*                             *
*    TABLE OF LN# XREFS       *
*                             *
*******************************

      4
6  60  70  78  92  108  126  136  142
148  154  168  206  208  214  216  240

      6
32  40  42  54  54  58  156  158

      8
12  46

     10
48

     12
16  56  142

     14
140

     18
4  6  10  12  32  40  150  160  176
178  190  210  246  250

     20
116

     26
20

     28
46  50

     34
50

     44
72  236  248

     56
52  54

     64
66

     66
64

     68
70

     70
68

     72
96  104  236  248
```

74
72 104

76
72 72

78
78

82
80

84
82 82 114 114 116 116

86
82

90
92

92
90

94
130 236 250

98
94 96

100
102

102
100

106
108

108
106

116
110

120
122

122
120

124
126

126
124

128
236 250

130
134

132
128 130

134
134

136
136

138
128

146
148

148
146

150
236

152
152

154
150 152

156
58 60 84

158
58 60 84

160
142

162
220

164
164

166
166

168
150 162 164

170
168 200 208

172
62 66 72 88 94 104 118 122 128
134 144 150 162 164 202 220 220
222 224

174
64 68 90 120 124 146 174

176
66 70 92 122 126 148

180
62 72 88 94 104 118 128 144
150 162 218

182
182

184
78 100 106 134 136 152 164 166
184 224 226

186
172 174 174

188
174 184 184 186 192

190
78 102 108 134 136 152 164 166
224 226

192
174 184

194
196

196
218

198
70 92 126 148

200
62 76 86 88 98 104 118 132 144
150 162 220

202
200 206 206 206

204
200 208 220 240

206
200

208
180

210
206

212
1 228

218
216

224
224

226
226

228
228 228 236

230
2 236 240

238
236 240

240
230

244
4 248 248 250 250

246
236

248
246

1 FEB 85 Version JRC/DCD

END OF LINE LIST

```
******************************
*                            *
*   POSTPROCESS.ELASTICITY   *
*                            *
*     TABLE OF LN# XREFS     *
*                            *
******************************
```

1
212

4
28 46 50 54 66 68 72 86 96
106 106 118 124 128 132 138 146
182 184 192 194 216

6
88

10
4 18 100 154 154 170 186

12
32 34

18
34 78

26
30 30 56 56 74 74 86 86 212
228

28
28

30
30

32
30

40
42

42
40

44
46

46
44

48
38

52
56 212 212 228 228

54
54

58
56

64
66

66
64

68
62

70
74 212 228

72
72

76
74

84
80

86
86

88
86

94
96

96
94

98
92

100
106 212

106
106

108
106

110
106

112
106

114
212 228

116
116

118
118

122
120

124
122

126
128

128
126

130
212 228

132
132

136
138

138
136

140
198

142
142

144
144

146
26 52 70 102 114 116 130 140
142

148
146 176 184

150
26 38 42 52 62 70 92 102 114
116 124 130 134 140 142 178 198
198 222

```
    152
40  44  64  94  126  136

    154
42  46  66  96  128  138

    158
26  38  52  62  70  92  100  114  130
140  196  222

    160
160

    162
28  54  72  106  116  118  132  142
144  224

    164
152

    166
152  152  162  162  164  168

    168
152  162

    170
28  54  72  106  116  118  132  142
144  224

    172
174

    174
196

    176
26  38  52  62  70  92  102  114  124
130  140  222

    178
176  182  182  182

    180
176  184  214

    182
176

    184
158

    186
182

    188
1

    196
194

    208
2  212  214  226  228  228  228

    214
212  212  212  216

    216
208

    220
4  86  228

    222
202

    224
224

    226
208  212

    228
226  228

1 FEB 85 Version JRC/DCD

END OF LINE LIST


*******************************
*                             *
*      PLOT.ELASTICITY        *
*                             *
*   TABLE OF LN# XREFS        *
*                             *
*******************************


     4
108  112  116  134  170  174  200  224
228  230  248  278  280  288  292  320
350  352

     6
44  48  160

     8
206  216  222

    10
12  18  48  48  98  98  156  156  160
160  164  164  166  168  198  220

    12
12  12  204

    14
20  22

    16
16  18  212

    20
96

    22
96
```

24
14 14 20 26 40 50 96 96 118
118 120 120 134 136 138 142 144
152 162 170 176 176 178 178 178
184 210 226 278 282

26
14 38 42

32
28

34
14 42 42

36
42 42

38
86

42
124 130

44
46 136 138 220

48
12 18 48 220

52
96 134 142 144 146 176 180 184
192 194 206 210 216 218 220 222
224

54
170 172 194 202 226

56
192

58
58

64
60

66
62

68
60 64

72
68

78
56 56 68

80
186

84
100 100 102 104

86
190

92
190 190

94
186

100
192

106
324

108
108

110
348

112
112

114
340

116
116

118
120 340

120
120 120 120

122
186 188

126
186 188

132
150

134
138 140 178

142
142 144 178

144
144

150
152

154
142 142 144 144

158
140 140

160
140 142 160

162
88 186 192

164
162

166
90 90 214 214

168
86 88 94 96

170
184 324 340 348

176
176 178

180
180

182
178 180

184
172 178

186
172 178 180

194
208 210 218

200
196

202
196 206

206
198

208
194 200 206

210
196 210 212 212

218
210 210 212 216 218 218 218 222
222 226

224
192 218 218 218 220

226
226 228 228 230

228
228

232
186

236
140 172 306

238
304

240
240

242
242

244
244

246
246

248
106 110 114 238 240 242 244

250
248 272

252
106 110 114 238 240 242 244 274
304 304 304 304

254
106 110 114 238 294

256
256

258
108 112 116 240 242 244 246 258

260
252

262
258 260 266

264
108 112 116 240 242 244 246

266
258

268
270

270
294

272
106 110 114 238 350

274
272 278 278

276
272 280 304 318

278
272

280
254

282
278

284
1

294
292

314
2 318 324 324 326 330 340 342
348 348 350 352 356 358 360 360

318
316 320

324
316 316

326
316 316 326 330 330 332

328
316 332 336 340

332
328

334
330 336 338

338
334

340
334 336

342
316 344

344
346

348
344

350
316 356 356 360 360

352
350

354
352 360 362

356
354

358
358

362
274

364
196

368
196

1 FEB 85 Version JRC/DCD

END OF LINE LIST

```
******************************
*                            *
* DISKETTE PREP. ELASTICITY  *
*                            *
*     TABLE OF LN# XREFS     *
*                            *
******************************
```

2
8 8 34

34
6

36
34

38
36

40
2 2

42
4

44
42

46
42

48
44 46

50
42

1 FEB 85 Version JRC/DCD

END OF LINE LIST

```
******************************
*                            *
*      TEXT.ELASTICITY       *
*                            *
*    TABLE OF LN# XREFS      *
*                            *
******************************


   500
10 10 12 12 14 18

   502
502

   504
502

   506
10 10 12 12 14 18

   508
6 6 8

   510
6 500

1 FEB 85 Version JRC/DCD

END OF LINE LIST
```

4.0 PROGRAMMER'S GUIDE

We have reserved a discussion of the programming details for this last section. In the earlier sections we provided the details required by a user of the program. If you are interested in merely getting a glimpse of the basic methodology of the finite element method, we have provided several examples with sufficient details to guide you quickly through the procedure. If you wish to review the computational details of the method, we have provided easy access to the intermediate calculations. The availability of the commented source code makes this kind of detailed exploration far more concrete and meaningful than would be possible otherwise.

4.1 Applesoft Code

Finally, we wish to provide sufficient details for those who might want to modify the code in order to deal with larger problems, to make the program easier to use, or to extend the code in ways which we have not anticipated.

These programs have already been used in a research setting, despite the limited memory of the Apple II.

R.E. Pitt and D.C. Davis have explored the influence of plant cell turgidity upon the apparent stiffness of potato tubers using the elasticity program (1984, 1976-1983). X. Yang and L.D. Albright (1985) have used the heat transfer program to explore the energy savings in bottom heating systems for greenhouse plants. The illustrative example used in Chapter 2 is a simplified version of their model.

Table 4.1 shows the memory requirements for each segment.

The table also shows the various programs and the number of diskette sectors used by each. In order to fit each program on a single diskette, we excised the comments and, whenever possible, placed common portions of code in subroutines. Only after we reclaimed the sectors on track two unused by DOS, did the programs fit. We reserved four sectors for a user-supplied hi-res screen dump utility at ($1D00-1FFF). The dummy routine provided consists of a single return statement.

Table 4.2 gives a memory map of the various program components. Note the placement of the utilities and the bank switched memory in high memory, i.e., different programs reside at the same addresses but on different banks of memory.

We have fully utilized the resources of the Apple II in the development of these programs. Whenever we needed graphics support, we located the Applesoft programs ($4000.BDFF) above the hi-res graphics but below the vectors to the DOS which we moved to the RAM card (table 4.2). The two program segments (SOLVE and POSTPROCESS) which require the largest data arrays do not make use of graphics. Consequently, we have lowered the Applesoft program in memory ($193E.BDFF) to just above the space reserved for the transient machine language utilities. We have kept the code for these segments small in order to enlarge the usable data space. This increases

Table 4.1 Program Characteristics

Applesoft Program	Sectors	Heat Bytes	Data Bytes	Sectors	Elasticity Bytes	Data Bytes
HELLO	11	2,395	29,860	11	2,394	29,861
GEOMETRY	79	19,791	12,464	79	19,791	12,464
GRID	83	20,965	11,290	83	20,965	11,290
PREPROCESS	83	20,740	11,515	72	18,043	14,212
SOLVE	48	12,010	30,137	50	12,511	29,666
POSTPROCESS	43	10,735	31,442	49	12,078	30,097
PLOT	82	20,361	11,894	88	22,154	10,101
DISKETTE PREPARATION	14	3,320	28,935	15	3,401	28,854

Table 4.2 Memory Usage

$300.3CF	Unused
$800.141F	FEM-C utilities
$1420.193D	Transient
$193E.1CFF	Shapes*
$1D00.1FFF	Hi-Res Dump (user supplied)*
$2000.3FFF	Hi-Res Screen*
$4000.BDFF	Applesoft Program and Data*
$BE00.BFFF	Vectors to DOS
$D000.FFFF ROM:	(Bank-switched memory) $D000.FFFF Applesoft and Monitor
RAM Card Bank 2:	$D000.FFFF FEM-A utilities, modified DOS
RAM Card Bank 1:	$D000.DFFF FEM-B utilities

* $193E.BDFF is used for program and data with the non-graphics, computation-bound SOLVE and POSTPROCESS

the data space from about 10,000 to about 30,000 bytes.

The Applesoft code has a common organization throughout all the segments. Line 1 is the cold start entry which immediately branches to an initialization section near the end of the program. Line 2 is always reserved as the warm start entry; this line branches to the end of the cold start initialization and presents the main menu which controls the main branching for the program. We placed the most speed sensitive code near the beginning of the program to speed up the interpreted BASIC. Also to reduce the time spent searching for program lines, we placed multiple statements on each line. This code compression will make editing difficult without an aid such as the Global Program Line Editor (Konzen 1982), a splendid piece of software. You will need extra print statements to replace the unprintable, embedded carriage returns shown as [M] in the commented source code unless you use such an editor. We reserved the odd line numbers (except line 1) to facilitate user modifications. Also, beware of renumbering the lines because longer numbers may cause parts of some statements to be lost.

We have retained a common definition of variables throughout the program segments wherever possible and used mnemonic codes whenever possible. We attempted to preserve consistency throughout. We standardized error handling, saving and retrieving data, menu organization, etc. Except in the cold start section, we dynamically removed arrays (if present) before dimensioning them in order to preserve the maximum flexibility in restart capability.

If the help messages and other aids fail to resolve a deadlock, simply restart by pressing control-Q from an input command to get a warm start. If that fails, simply press 'reset' or 'control-reset' and issue the 'RUN' command. With the program segmentation and frequent data storage, you can recover from even serious blunders.

We cannot leave this discussion without some comment about the project. The most obvious factor is the presence of the Applesoft Basic Interpreter in every Apple II. This was readily available and, therefore, user-familiarity was obvious.

Interpreted code is slower than compiled code. However, the interpreted code is significantly more compact and memory is at a premium. We have compensated for the slower execution speed of interpreted code by the use of extensive 6502 assembly language code, which is described in the next section.

Finally, you may solve slightly larger problems by deleting temporarily unneeded portions of code. For example, in PLOT (Heat) you may remove the picture retrieval option (lines 308-320) or you may disable zoom plots (lines 110-134). Although you can enlarge the problem size slightly, we believe that significantly larger research problems are beyond the scope of this instructional package. We hope this educational tool will facilitate, not prevent, your transition to the mainframe programs.

4.2 Assembly Language Utilities

While the code of an interpreted language such as Applesoft is compact, execution speed is slower. Assembly language, on the other hand, provides much faster execution and access to additional capabilities. In the following section we describe the enhancements which have been added and used by the these programs. Many of these extensions are enhancements of published assembly language programs. We note sources for each of the commands, which are presented in alphabetical order.

4.2.1 Description of Commands

1. Applesoft RAM Utilization

Usage: &A()

Use as a direct or indirect statement with no parameters within the parenthesis or use control-A, if entered as the first character of an input statement immediately following the default input which is delineated by the square brackets [].

This command is especially useful in determining the size of problems being solved, especially as you approach the maximum memory utilization. The beginning, length, and end of each section of memory is given in both hexadecimal and decimal notation. The categories reported are program, variables, arrays, free space, and strings and are denoted by the first letter of the category name (Golding 1981b).

2. Beep prompt

Usage: &B(P%,D%) and &B(P%,D%,N%)

Use with parameters P%, the period (0-255); D%, the duration (0-255); and N%, the optional repetition factor (0-255), if omitted defaults to 1. Beeps of different pitch are set by P%. You can use this command to produce a timed pause if the period P% = 0. You can increase the pause duration by increasing the number of repetitions N% and the duration P% of each silent pause (Irwin, 1982).

3. Clear Arrays

Usage: %C(A,B,...)

This dynamically recovers memory reserved for the listed arrays by previous DIM dimension statements. With one command you can clear an arbitrary number of arrays separated by commas. The dimension subscripts are not specified but are determined internally. The variable cross-reference listing includes these array references as simple variables and can be misleading.

This utility not only allows more efficient utilization of the limited memory but also allows more flexible programming. By preceding each dimension statement with a clear array, we made possible more flexible re-starting and re-entry conditions. We modified the code to permit non-existent arrays to be 'cleared' without producing an error (Francios 1983, 173-179).

4. Display screen selection

Usage: &D(T) or &D(G) or &D(V)

These commands set the display to text page one, to mixed graphics with four text lines at the bottom, or to full view of page one high resolution graphics without erasing the contents of either text or graphics. This command also sets a flag to indicate which screen configuration is currently being displayed. You must set the flag immediately after each TEXT or HGR command but it will otherwise remain current. You can also issue this command interactively by typing control-T, control-G, or control-V whenever the &I( ) command described below is awaiting input (as indicated by the square brackets enclosing the default input).

5. Fill a closed region (shading)

Usage: &F(CO%,OP%,X,Y)

This fill routine requires that you bound the region to be filled by a curve generated using HCOLOR = 3. The first and second parameters specify the color and fill options (0-16); we have adjusted the color parameter for a monochrome monitor. The X and Y values correspond to arbitrary starting coordinates within the region to be filled, usually taken to be the centroid (McBride, 1983).

6. Input a variable

Usage: &I(P$;DF$;A$)
&I(P$;DF$;A$,H$)
&I(P$;DF$;A$,H$,BC%,L%,PC$)

This utility is the centerpiece for the entire program. We will describe various other usage forms after we review these three. You can give the input prompt any expression appropriate for a print statement. This includes literals such as a string constant enclosed in quotes. The prompt may be separated from the second parameter by either a semicolon or a comma. A semicolon suppresses the carriage return while a comma produces a return after the prompt is printed.

The second positional parameter is the default answer DF$ that will be displayed in inverse video and enclosed by square brackets. The default may be either a variable expression such as would be acceptable in a normal PRINT statement or an unchanging literal. However, since the default value will be assigned to the third parameter A$, a single input variable, the type of the default and input variable must agree. We show only the string input version with A$, but the default and variables must be alike (i.e., may be both real, integer, or string types or can be either simple or array element type). The separator between the default and the input variable may be a comma (for a carriage return) or a semicolon for the cursor to immediately follow the default. All positional separators after the first two must be commas.

We have described the simplest format; but in order to support the help messages (see sections 2.3 and 3.3), we have always used the optional fourth parameter. The help parameter H$ must be either a string literal or a string variable.

If you type a question mark '?' as the first and only character for an input request following the square brackets of the default and press 'return', you will be given a code referring you to the appropriate help message. Following the word 'HELP' a number will be printed to identify the program segment. The integer part of the number (0-255) retrieved from memory location 2164 ($0874) indicates the program number. Following the period is the help string from the input parameter list. We have used a string of sequential numbers but text could have been used to identify the message.

Use a right parenthesis to terminate the parameter list after the third or subsequent parameters. You must preserve the order of the parameters included. If you do not wish to provide a help message, you need a place holder, a null string. Then if you request help, a 'NO HELP OPTION' message will be printed.

The fifth parameter is an integer variable. We have used BC% throughout to denote this branch code. If you enter '<' as the first and only character before pressing return, a value of 1 will be assigned to the integer variable BC%. The Applesoft code then directs a logical branch backward in the code. This command allows you to return to previous questions for corrections.

If you enter '>', a value of two will be assigned to the integer variable. This allows the Applesoft code to branch forward, accepting defaults and skipping questions which you do not wish to answer. Because a forward branch is not always appropriate, you may get some unexpected results. A third code ':' is user definable and was used for identifying previously entered points in GEOMETRY.

The sixth parameter allows you to restrict the number of keystrokes acceptable in the answer. The most common example is the restriction on the number of characters allowed in a file name. The parameter may be a numerical constant (0-230) or an integer variable. The specified maximum number of characters is depicted with underscores. As you enter characters, the underscores are replaced. If you use the left and right arrows, the line is restored. Unlike the normal Applesoft input, deleted characters disappear. Control-X causes the entire line to be restored to underscores. A length of zero (L% = 0) disables the length restriction.

Finally, you may include the seventh parameters to restrict the input response to an acceptable range. This parameter may be any logical test which would be acceptable in the Applesoft IF statement. Only responses that meet this test, including the default, will be accepted. Frequent uses are the restriction on menu option numbers and the range of input coordinates. If you are unable to get the input command to accept your response, you may examine the restriction by pressing control-R as the first input character (without return).

In addition to the one character commands which require a return ('?' '<', '>') and the one non-printing command not requiring a return (left and right arrow control-X, and control-R), there are several other non-printing commands. You must enter these control character commands as the first character (except control X), because they are otherwise excluded from the input. The warm restart command to quit the current step and to return to the principal menu of this program segment is the most powerful and most frequently used. Line 2 of each program segment (control-Q) has been reserved as the warm restart vector. To implement this option we removed any pending GOSUB return addresses from the stack. We either placed all dimension statements in the cold start initialization section, or we cleared the array before redimensioning (see &C()).

You may monitor the memory usage with control-A. This activates the &A() command described above to indicate the current memory usage. Remember that the memory usage is dynamic. With problems that tax the memory limits, this command can help guide your mesh refinement efforts.

You can switch the video display from the input command with control-T, control-G, and control-V to display the text, mixed graphics, or full view graphics. Note the keyboard location of these commands as a convenient reminder of their usage. Control- L and control-O turn the listing control on or off. You may also use control-Z to turn the listing control off. Use of these can be awkward because the listing control must be turned off when DOS commands are executed. See &L() for a full description.

You control the important screen dump facility by pressing control-P. If you have a printer interface in slot 1 which accepts output at $C102, you may use control-P to print the text screen whenever the full text screen is in view. If you have installed a hi-res screen dump driver at $1D00.1FFF and have the graphics printer switches properly set, you may print the hi-res pictures when the hi-res picture is displayed. Control-T, control-G, and control-V may be used to choose the screen to be printed. See discussion of the &P() command for additional details. The string input routine allows shorter input strings to reuse existing memory allocations (Nacom 1983).

7. List control for the video display

Usage: &L(P%) where the argument may be a constant or a variable having values 0-11 or 255.

You can control the rate of presentation of the data on the screen in several ways. We used &L(10) in the program to control the output with paging. When the screen fills, the program pauses without a message until you press return. You may press the space bar to scroll a line of data or press one of the numeric keys (0-9) to set the delay level for continuous scrolling. (0 = no delay; 9 = maximum delay). You may change this 'on-the-fly' or press the space bar to pause and then revert to line-by-line listing (space) or to page-by-page listing (return).

If you have a paddle installed, you may press 'P' to control the scrolling with the paddle or joystick. A single press of the button on the paddle or joystick halts the listing; a second press resumes the listing. If the printer will accept output as described in the &I() command, the text screen may be printed by pressing control-P whenever the scrolling has been stopped (Black 1981, 59; Hartley 1981, 71).

8. Matrix operations

```
Usage: &M(A = (expr))
       &M(A = IDN(1))
       &M(A = PNT(9))
       &M(A = TRN(B))
       &M(A = B.C)
       &M(A = MIN(B))
       &M(A = MAX(B))
       &M(A = ABM(B))
```

This command handles various matrix operations for arrays of real numbers. It evaluates every element, except row zero and column zero for a previously dimensioned array denoted here by 'A'. It evaluates and assigns to each element of the matrix an expression (expr) enclosed in parentheses. You may create an identity matrix with the IDN(1) command and print an array with PNT(9). The nine sets the number of digits allowed. PNT(0) allows the field width to automatically print the full precision. You can obtain row and column interchange with the transpose command TRN(B). Use a period to denote matrix multiplication. Obtain the minimum, maximum, and maximum of absolute values for each column by the last three commands (Bongers 1981a, 106-110; Bongers 1981b, 31-47).

9. Names of picture files

```
Usage: &N(N$,BR,P)
       &N(N$,BR,P,NR%)
       &N(N$,BR,P/T/A,NR%,SF%)
```

You can read the names of files from the data diskette directory with this command which may have 3, 4, or 5 parameters. The first parameter is a single subscript string array previously dimensioned to receive the file names. BR assigns a name to the beginning row BR of the array (usually row 1). The third parameter may have one of three values: P to retrieve all file names which end with '.PIC' denoting a picture file; T to retrieve all filenames which end with '.TXT' denoting text files such as FILEINFO.TXT; and A to indicate that all file names are to be retrieved. The 'T' parameter would allow the programs to be generalized such that several 'FILEINFO' files (and hence several different problems) could be placed on the same data diskette. At present, only one directory file named 'FILEINFO.TXT' may reside on the diskette. If this file name were preceded by the keyword, several could co-exist on the diskette (Conrad 1984, 149; Freeman 1982, 8).

10. Order the elements of an array

Usage: &O(A,BR TO ER,COL,A/D)

This command orders (or sorts) the rows of a one or two dimensional array from the beginning row (BR) to the ending row (ER), where BR and ER may be a subset of the actual rows. The sorting is based on the values in column COL. You may return the resulting array in ascending (A) or descending (D) order. This uses the quicksort algorithm (Bongers 1984, 33-53).

11. Print the screen contents

Usage: &P()

The print command requires parentheses but no argument. If your printer responds as described in the &I() command, then this command will dump the text or graphics screen(s) displayed just as you can from within an input command with control-P. See the end of this section for modifications for other printers (Mottola 1980b, 40).

12. Recall an array

Usage: &R(A,N$)

You may obtain binary data storage and retrieval by saving an image of the array A in memory rather than using the customary process of converting the values to an ASCII text file. You must correctly dimension A before using it. The N$ variable specifies the name of the file to be retrieved (see Store) (Bongers 1982a, 121).

13. Store an array

Usage: &S(A,N$)

This command rapidly saves the array A as a binary file with name N$. Since the array for retrieval must match the original array, Applesoft code is used to save dimensions of the array in 'FILEINFO.TXT' (Bongers 1982a, 121).

14. Determine WHICH rows of an array have the elements of a specified column within a specified range.

Usage:
&W((A,BR TO ER,COL),LL,UL,NR%,LR%,UR%)

This command allows you to locate the rows of a one or two dimensional array which have values of a specified column within a prescribed range. Because memory is so limiting you sort the array A (real or integer) 'in-place' using &O. Consequently, the array A, the beginning row BR, ending row ER, and column COL parameters are the same as in the ordering (sorting) command &0. The next two parameters give the lower and upper limits (inclusive) for the values to be bracketed. The final three parameters return the desired information. NR% is the number of rows that have values bounded by the lower and upper limits. If this value is greater than zero, then LR% gives the lower row number and UR% gives the upper row number of the sorted array A which satisfy the bounds. Otherwise, beware; if NR% = 0, the bounds are not meaningful.

DOS MODIFICATIONS

In addition to the above fourteen extensions to Applesoft, we have employed several extensions to the Disk Operating System.

15. Relocated DOS

We have relocated DOS from its normal place in RAM to the RAM card. This makes 10,000 additional bytes of data space available and, thereby, greatly extends the size of problem which can be solved (Bongers 1981c; Konzen 1982).

16. Faster Loading of Files

Because the Applesoft code is so large, the time required to load each program segment becomes appreciable. This modification decreases the loading time by a factor of 3 to 5 (Burt 1983, 48-59).

17. Listing of Text Files

Usage: TDUMP filename,S6,D1

We replaced the 'Chain' command used with Integer Basic with 'TDUMP' which lists sequential and random access text files. This command is frequently useful for examining the directory file 'FILEINFO.TXT' (Bongers 1982b).

18. Preparation of data diskettes

Usage: PREP

We made possible a substantial increase in data storage on a diskette by the removal of DOS and the greeting program. We reserved only sector 0 of track 0 and all of track $11. This makes 47 additional sectors available for data storage. This command replaces the standard INIT command and may include the slot and drive parameters (e.g., PREP,S6,D2). (Anderson 1983)

19. Verify the existence of files.

We have changed the name of the 'Verify' command to 'CKFILE'. Yet to be implemented is the ability to simply verify that a file name exists on the diskette without the delay associated with reading the entire file.

20. Other assembly language utilities

Because the hi-res screen divides the user RAM, we decided to use the memory below the graphics screen and above the text screen ($801.1FFF) for assembly language utilities. We have used the Ampersand command structure presented by Paul Irwin (1982, 19-37) to access the various utilities. Because even greater space was needed, we used bank 1 of the language card to store several of the utilities until needed. We used the

RAMCARD loader presented by Sandy Mossberg (1983, 167) to move the utilities to the bank-switched area. We used a portion of the RAM below the hi-res screen as a transient area to which the utilities could be moved for execution. The vectoring for access to DOS and Applesoft routines in the same address range but on different banks of memory would have become excessive if the routines had been left resident on bank 1.

The fill routine, the fast load enhancement, the text file lister, and the data diskette initializing programs reside on bank 2 below DOS with maxfiles limited to two.

The page 3 RAM ($300.3CF) has not been used and is available for user-defined utilities.

We found the *Nibble, Call Apple* and *Micro* magazines to be valuable resources in the development of these programs. Of the various software tools used in the development of these programs, we found the following to be of noteworthy utility: the Global Program Line Editor, the Big Mac.LC assembler, the Lister, the Comparer, VARDOC, LINEDOC, and Bugbyter.

4.2.2 Acknowledgments

We are pleased to acknowledge permission to utilize all or parts of the following **copyrighted** assembly language utilities:

1. *Nibble* magazine

Sandy Mossberg wrote the routine used to move the code to bank one of the RAM card for on-line storage (Mossberg 1983).

Paul Irwin developed the structure for loading the Applesoft program above the hi-resolution graphics page and for routing the ampersand command extensions to Applesoft according to the first letter following the ampersand (Irwin 1982). We extended the beep routine presented by Irwin to include the repetition parameter and the timed pause, which can be aborted by pressing the space bar.

We handled text screen dumps using a program by Mottola (1980b); we added access to the routine through control-P from an input command and by &P() as an Applesoft command.

As a faster alternative to sequential text files for storage of data, the program stores and retrieves array images as binary files (Bongers 1982a).

With the ability to delete arrays which were no longer needed we increased the size of problems which could be solved (Francios 1983). In order to retain the restart capability while still delaying array dimensioning until an array was needed, we routinely deleted arrays before they were dimensioned; we added a provision for 'deleting' non-existent arrays.

Although we used program compression techniques, the Applesoft program segments are still quite large (see Table 4.1). We obtained a three to five fold increase in loading speed using the 'fast load' enhancement of Burt (1983) adapted for DOS relocated to the RAM card.

The addition of a catalog reader which placed the names of picture files into an array enhanced the process of picture retrieval (Conrad 1984). We generalized this routine to work with the relocated DOS and to return the names of all files or only those with names ending with '.PIC' or with '.TXT'.

The graphics interface for user input required extensive sorting of arrays. In fact, the computation time for this aspect exceeded the computation time for solving the system of equations! The quicksort utilities of Cornelis Bongers (1984) significantly increased the speed of response. We extended this utility to select the rows of an array for which a range restriction on a specified column had been designated; in order to avoid the creation of an additional array, the results were returned as the beginning and ending subscripts for the sorted array.

2. *Call A.P.P.L.E.*

We used the closed curve filler of McBride (1983) to provide shading as a means of displaying magnitudes of a quantity in a region -- both for input and output arrays. We relocated this utility to operate on bank 2 below the relocated DOS.

We substantially extended the free space and pointers utility of Golding (1981b) to show memory utilization in both decimal and hexadecimal formats, giving both the beginning, ending, and length values. We provided access to this utility through the input command and through an Applesoft command.

The massive input routine includes the memory re-use technique for strings presented by Nacom (1983); we added protection against source code modification.

The diskette catalog retrieval function includes a 'free sectors' function adapted from Pump et al. (1982).

We included a text file dump program (Bongers 1982b) to permit you to review the status of the data files as recorded in the FILEINFO.TXT random access directory file.

We used the DOS relocating utility of Cornelis Bongers (1981c) to move DOS 3.3 from its normal location to bank 2 of the RAM card, freeing slightly more than 10,000 bytes of memory in the contiguous block used

by the Applesoft program.

3. *Micro* magazine

We used the list controller of Preston Black (1981) to control the listing of tabular results. We added extensions to permit screen dumps while paused and to set scroll speed with listing in progress. We also added a command to enable/disable this listing control, a feature needed to bypass DOS commands. You can enable this feature with control modes of a line-at-a-time, page-at-a-time, or continuous scrolling at a specified rate.

We used another utility by Bongers (1981a) for matrix operations. We did not include the matrix inversion command because the compact storage used for the banded, symmetric stiffness matrix could not be preserved.

4. *Apple Orchard* magazine

We utilized the diskette initialization code presented by Anderson (1983). When preparing a data diskette, rather than preparing an initialized master data diskette which must be duplicated, we replaced the INIT command in DOS with a PREP command that created a DOS-less diskette which used only sector 0 of track 0 of the 48 sectors normally reserved for DOS.

5.0 REFERENCES

GENERAL REFERENCES

Abramowitz, Milton, and Irene Stegun. 1965. *Handbook of Mathematical Functions.* Washington, DC: U.S. Government Printing Office.

Anderson, William J. 1983. *MSC/NASTRAN Interactive Training*, Program. New York: John Wiley and Sons.

Batchelor, G.K. 1967. *An Introduction to Fluid Dynamics.* Cambridge: Cambridge University Press.

Bathe, Klaus-Jurgen, and Edward L. Wilson. 1976. *Numerical Methods in Finite Element Analysis.* Englewood Cliffs, NJ: Prentice-Hall, Inc.

Byerly, William Elwood. 1959. *An Elementary Treatise on Fourier's Series.* New York: Dover Publications.

Chapman, D.C., R.H. Rand, and J.R. Cooke. 1977. A Hydrodynamical Model of Bordered Pits in Tracheids. *J. Theoretical Biology.* 67:11-24.

Collins, R.J. 1973. Bandwidth Reduction by Automatic Renumbering. *International Journal for Numerical Methods in Engineering.* Vol. 6:345-356.

Cook, Robert D. 1981. *Concepts and Applications of Finite Element Analysis.* 2nd ed. New York: John Wiley and Sons.

Cooke, J.R., and D.C. Davis. 1983. Microcomputer-based Instruction in Finite Element Analysis: Part 2 - Applications. Paper 83-5538. American Society of Agricultural Engineers, St. Joseph, MI 49085.

Cooke, J.R., and D.R. Price. 1969. Analysis of Subsurface Electric Heating. Paper No. 69-848. American Society of Agricultural Engineers, St. Joseph, MI 49085.

Davis, D.C., and J.R. Cooke. 1983. Microcomputer-based Instruction in Finite-Element Analysis. Part 1- Implementation. Paper 83-5538. American Society of Agricultural Engineers, St. Joseph MI 49085.

Desai, Chandrakant S., and John F. Abel. 1972. *Introduction to the Finite Element Method. A Numerical Method for Engineering Analysis.* New York: vanNostrand Reinhold Co.

Gallagher, R.H. 1975. *Finite Element Analysis Fundamentals.* Englewood Cliffs, NJ: Prentice-Hall, Inc.

Gustafson, R.J. 1977. Finite Element Method Package of Learner Programs (FEMPAC). University Computation Center Technical Report UCTR77001. University of Minnesota, St. Paul.

Huebner, Kenneth H. 1975. *The Finite Element Method for Engineers.* New York: John Wiley and Sons.

Kirkham, Don, and W.L. Powers. 1972. *Advanced Soil Physics.* New York: Wiley-Interscience.

Kober, H. 1957. *Dictionary of Conformal Representations.* New York: Dover Publications.

Krieth, Frank. 1958. *Principles of Heat Transfer.* Scranton, PA: International Textbook Co.

Lamb, Horace. 1945. *Hydrodynamics.* 6th ed. New York: Dover Publications.

Love, A.E.H. 1944. *A Treatise on the Mathematical Theory of Elasticity.* 4th ed. New York: Dover Publications.

Martin, Harold C., and Graham F. Carey. 1973. *Introduction to Finite Element Analysis: Theory and Application.* New York: McGraw-Hill Book Company.

Milne-Thomson, L.M. 1968. *Theoretical Hydrodynamics.* 5th ed. New York: Dover Publications.

Moon, Parry, and Domina Eberle Spencer. 1961. *Field Theory for Engineers.* Princeton: D. van Nostrand Company, Inc.

Moon, P., and D.E. Spencer. 1971. *Field Theory Handbook.* 2nd ed. New York: Springer-Verlag.

Oden, J.T., and H.J. Branchli. 1971. On the Calculation of Consistent Stress Distributions in Finite Element Approximations. *International Journal for Numerical Methods for Engineering.* 3:317-325.

Polubarenova-Kochina, P.YA. 1962. *Theory of Groundwater Movement.* Princeton: Princeton Univ. Press.

Rothe, R., F. Ollendorff, and K. Pohlhausen. 1933. *Theory of Functions as Applied to Engineering Problems.* New York: Dover Publications.

Segerlind, Larry J. 1976. *Applied Finite Element Analysis.* New York: John Wiley and Sons.

Segerlind, Larry J. 1984. *Applied Finite Element Analysis.* 2nd ed. New York: John Wiley and Sons.

Scott, N.R., J.R. Cooke, R.H. Rand, R.A. Parsons, and K.D. Mahan. 1983. Analysis of Earth-Air Heat Exchange. Lawrence Berkeley Laboratory, Berkeley, CA.

Smythe, William R. 1950. *Static and Dynamic Elasticity.* New York: McGraw-Hill.

Steinmueller, G. 1974. Restrictions in the Application of Automatic Mesh Generation Schemes by Isoparametric Coordinates. *International Journal for Numerical Methods in Engineering.* 8:289-294.

Stroshine, Richard. 1984. Letter to author, 6 July 1984.

Timoshenko, S., and J.N. Goodier. 1951. *Theory of Elasticity.* 2nd ed. New York: McGraw-Hill Book Company.

Tranter, C.J. 1962. *Integral Transforms in Mathematical Physics.* New York: John Wiley and Sons.

Walker, Miles. 1964. *The Schwartz-Christoffel Transformation and Its Application - A Simple Exposition.* New York: Dover Publications.

Yang, X., and L.D. Albright. 1985. *Finite Element Analysis of Temperature Distributions in Bottom-heated Nursery Containers.* Acta Horticulturae (in press).

Zienkiewicz, O.C. 1977. *The Finite Element Method.* 3rd ed. New York: McGraw-Hill Book Company.

Zienkiewicz, O.C., and K. Morgan. 1983. *Finite Elements and Approximation.* New York: John Wiley and Sons.

ASSEMBLY LANGUAGE REFERENCES

Anderson, J. David. 1983. Alleviate the 'DISK FULL' Problem. *Apple Orchard.* (March):46-47.

Apple Computer, Inc. 1981. Apple II Monitors Peeled. Cupertino, CA.

Black, Preston R. 1981. List-Controller. *Micro,* no. 40 (September):59-61.

Bongers, C. 1981a. Applesoft and Matrices. *Micro,* no. 43 (December): 106-110.

Bongers, C. 1981b. In the Heart of Applesoft. Micro, no. 33 (February):31-47.

Bongers, C. 1981c. Loading DOS 3.3 on the Language Card. *Call APPLE,* (July/August):9-21.

Bongers, C. 1982a. Amper Disk Store and Recall. *Nibble* 2(6):121.

Bongers, C. 1982b. TDUMP with DOS in RAM Card. Call APPLE in Depth: All About DOS. Renton, WA: Apple Pugetsound Program Library Exchange.

Bongers, C. 1984. Quicksort II. *Nibble* 5(3):33-53.

Bredon, Glen E. 1981. Big Mac Macro-Assembler/TED. Call APPLE. Renton, WA: Apple Pugetsound Program Library.

Bredon, Glen E. 1982. Big Mac.LC Language Card Supplement. Renton, WA: Apple Pugetsound Program Library.

Burt, Tom. 1983. Fast Load Enhancement. *Nibble* 4(5):48-59.

Conrad, Bruce. 1984. Amper Catalog. *Nibble* 5(2):149.

Christ, Bill. 1983. The RAM Manager. *Nibble* 4(4):13-29.

Francios, P. 1983. Delete Arrays. *Nibble* 3(8):173-179.

Freeman, Larry. 1982. The Directory Getter. *The Abacus* 4(7-8):8.

Golding, Val J. (ed). 1981a. Call APPLE In Depth: All About Applesoft. Renton, WA: Apple Pugetsound Program Library Exchange.

Golding, Val J. 1981b. Free Space and Pointers. in: Call Apple In Depth I: All About Applesoft. Renton, WA: Apple Pugetsound Program Library Exchange.

Giambalvo, Vince. 1982. Print to Disk. *Call APPLE,* (August): 51-55.

Harris, Mark. 1982. New Uses for the RAM Card. *Call APPLE,* (December):25-29.

Hartley, Chuck. 1981. Apple-speed. *Nibble* 2(2):71.

Irwin, Paul. 1982. Amp-L-Soft Utilities. *Nibble* 3(7): 19-37.

Konzen, Neil. 1982. Global Program Line Editor. Apple Pugetsound Program Library Exchange. Renton, WA. (Now available from Beagle Bros Micro Software. San Diego, CA 92110)

Little, Gary B. 1983. Introductory RAMCard Programming. *Nibble* 4(8): 149-151.

Little, Tom. 1982. Ins and Outs. *Call APPLE,* (July):19-24.

Logan, Coleman. 1983. Switchcraft: Memory Management. *Call APPLE,* (March):9-12.

Luebbert, William F. 1982. *What's Where in the Apple.* Chelmsford, MA: Micro Inc.

McBride, James. 1983. Ap. Graphics: The Filler. Renton, WA: Apple Pugetsound Program Library Exchange.

Mossberg, Sandy. 1983. Disassembly Lines RAM Cards and ROM Writers. *Nibble* 4(2):167-174.

Mottola, R.M. 1980a. Applesoft Floating Point Routines. *Micro,* no. 27 (Aug):53-55.

Mottola, R.M. 1980b. Machine Language Screen Dump. *Nibble* 1(3):40. (Also: Nibble Express 1:18.)

Nacom, Bob. 1983. The Penultimate Penultimate Input-anything Routine. *Call APPLE,* (January):43-45

Pruez, Harry L. 1980. Apple II Floating Point Utility Routines. *Micro,* no. 22 (March):7-11.

Pump, Mark. 1981. DOS Internals: An Overview. *Call APPLE,* (February):8-12.

Pump, Mark, V. Golding, W. Reynolds III, and R. Welman. 1982. Free Space/File Dump Poker. in: Call APPLE in Depth: All About DOS. Renton, WA: Apple Pugetsound Program Libray Exchange.

Worth, D., and P. Lechner. 1981. *Beneath Apple DOS.* Resdin, CA: Quality Software.

HIGH RESOLUTION GRAPHICS SCREEN DUMP UTILITIES

Anderson, Terry L. 1981. Utilities for the Paper Tiger 460. *Micro,* no. 39 (August):44-51.

Delaney, Robert M. 1982. Hi-Res Dump 2X. *Nibble* 3(6):95-103.

Hudson, David. 1980. Enhanced Paper Tiger Graphics for the 460G. Computer Stations, Inc. Granite City, Il.

Little, Gary. 1983. Apple Graphics for Okidata Microline 80: Revisited. *Micro,* no. 60 (May):80-83.

Taylor, Charles F. Jr. 1982. Apple Graphics for Okidata Microline 80. *Micro,* no. 48 (May):48-53.

Walker, Robert D. 1984. Hi-Res Screen Dump for the Epson MX-80. *Micro,* no. 71 (April/May):55-61.

Wiley

ISBN 0-471-82337-6